



CUESTA
LAWYERS

Señora
Sandra Isabel Calderón Rodríguez
Directora de Nuevas Creaciones
Superintendencia de Industria y Comercio
E. _____ S. _____ D. _____

Referencia: Solicitud de patente PCT denominada “**PROCESAMIENTO DE LUMINANCIA CON ADAPTACIÓN DEL RANGO HDR EN COMPUTADORES**”

Asunto: Respuesta al oficio No. 11887 del 9 de junio de 2026

Expediente No. NC2026/0008439

Juan Carlos Cuesta Quintero, mayor de edad, identificado como aparece al pie de mi firma, obrando en mi calidad de apoderado de **KONINKLIJKE PHILIPS N.V.** doy respuesta al Oficio No. 11887 del 9 de junio de 2026, bajo los siguientes términos:

1. Descripción

El Examinador vierte que el capítulo descriptivo allegado en la solicitud presenta diferencias con la descripción de la solicitud internacional No. PCT/EP2024/083790, por lo tanto es necesario que se allegue el citado documento de forma completa. En concreto, esta Oficina evidencio que el capítulo descriptivo está incompleto desde el párrafo que comienza “DESCRIPCION DETALLADA DE LAS REALIZACIONES”.

Adjunto se remite el documento solicitado que como se puede apreciar, a partir de la página 113 incluye la “DESCRIPCIÓN DETALLADA DE LAS MODALIDADES”.

Entendiendo que los principios rectores del derecho administrativo, especialmente el de eficacia y el de economía procesal, dictan que las entidades estatales removerán impedimentos meramente formales y que buscarán que los trámites cumplan sus finalidades, solicito a esta Dirección, que:

- i. Dé por contestado el requerimiento de forma en referencia.
- ii. En caso de no encontrarse satisfecho con dicha respuesta, **se conceda una cita** para dar la claridad sobre la respuesta presentada.
- iii. Se abstenga de emitir declaratoria de abandono del trámite de concesión de una patente, teniendo en cuenta que mi representada ha mostrado evidente interés en se le otorgue el derecho de patente para la invención de la referencia, y ha sido diligente y cuidadosa al dar respuesta al presente requerimiento.

Cordialmente,

Juan Carlos Cuesta Quintero
C.C. 79.339.809 de Bogotá
T.P. 43.273 del C.S. de la Judicatura

PROCESAMIENTO DE LUMINANCIA CON ADAPTACIÓN DEL RANGO HDR
EN COMPUTADORES

CAMPO DE LA INVENCION

5

La invención se relaciona con coordinar el brillo de los píxeles en diversos elementos visuales para visualizar, en particular específicamente para entornos informáticos, y en particular donde un sistema de ventanas puede coordinar los diversos elementos visuales en un lienzo total que se cargará en un búfer de salida de escaneo de una unidad de procesamiento de gráficos (GPU, por sus siglas en inglés), antes de exhibirlos en una o más pantallas de visualización conectadas (p. ej., mediante el uso de extensión de pantalla, o duplicación de pantalla), cuyas pantallas de visualización pueden tener diferentes capacidades máximas de luminancia visualizable, y en particular si al menos una de las pantallas es una pantalla de alto rango dinámico.

20

ANTECEDENTES DE LA INVENCION

Durante más de medio siglo, una tecnología de representación/codificación de imágenes que ahora se denomina **bajo rango dinámico (LDR, por sus siglas en inglés) o rango dinámico estándar (SDR, por sus siglas en inglés)** funcionó

perfectamente para crear, comunicar y mostrar imágenes electrónicas como videos (es decir, secuencias de imágenes sucesivas en el tiempo). Colorimétricamente, es decir, en lo que respecta a la especificación de los colores de los
5 píxeles, se basaba en una tecnología que ya funcionaba bien décadas antes para materiales fotográficos y pinturas: bastaba con poder definir, y mostrar, la mayoría de los colores que se proyectaban desde un eje de colores acromáticos (también conocidos como grises) que abarcaba
10 desde el negro en el extremo inferior hasta el color acromático más brillante, dando al espectador la impresión de ser blanco. Para la comunicación televisiva, que se basa en un mecanismo aditivo de creación de color en la pantalla, era necesario comunicar un triplete de componentes de color
15 rojo, verde y azul para cada posición en la pantalla de visualización (píxel), ya que con un triplete con proporciones adecuadas (p. ej., 60 %, 30 %, 25 %) pueden crearse casi todos los colores, y en la práctica todos los colores necesarios (el blanco se obtiene llevando los tres
20 canales de visualización al máximo, con una señal de conducción $R_{\text{máx.}}=G_{\text{máx.}}=B_{\text{máx.}}$).

Los primeros estándares de televisión (NTSC, PAL) comunicaban los componentes de color como tres señales de voltaje (que definían la cantidad de un componente de
25 color entre 0 y 700 mV), donde las posiciones temporales

a lo largo de la señal de voltaje se correspondían usando una ruta de exploración con los píxeles en la pantalla.

Las señales de control generadas en el lado de creación instruían directamente sobre la proporción que debía mostrar la pantalla (aparte de una precorrección gamma fija accidental en el transmisor, debido a que la física del tubo de rayos catódicos tomaba aproximadamente una potencia al cuadrado del voltaje de entrada, lo que habría hecho que los colores oscuros fueran mucho más negros de lo que se pretendía, p. ej., como los ve una cámara) en el lado de creación. Así, un color de 60 %, 30 % y 25 % (que es un rojo oscuro) en la escena capturada, se vería sustancialmente similar en la pantalla, ya que se regeneraría como un color de 60 %, 30 % y 25 % (tener en cuenta que el brillo absoluto no importaba mucho, ya que el ojo del espectador se adaptaría al blanco y al brillo promedio de los colores que se muestran en la pantalla). Esto puede denominarse “conexión directa”, sin procesamiento de color adicional (excepto por el procesamiento arbitrario e innecesario que un fabricante de pantallas aún podría realizar para, p. ej., hacer que su cielo parezca más azulado). Por razones de retrocompatibilidad con las antiguas emisiones de televisión en blanco y negro, en lugar de comunicar una señal de voltaje roja, verde y azul, se transmitían una señal de brillo y dos señales de diferencia de color

llamadas croma (en la nomenclatura actual, la croma azul Cb y la roja Cr). La relación entre RGB e YCbCr es sencilla: pueden calcularse entre sí usando una matriz fija simple de 3x3 (cuyos coeficientes dependen de los espectros de emisión de los tres colores primarios y están estandarizados, es decir, también pueden emularse electrónica e internamente mediante pantallas LCD que, de hecho, pueden tener características ópticas diferentes, de manera que, desde el punto de vista de la comunicación de imágenes, todas las pantallas SDR son iguales).

Estas señales de voltaje fueron posteriormente digitalizadas para la televisión digital (basada en MPEG, etc.), según lo prescrito en la norma Rec. 709, y se definen las distintas cantidades, p. ej., del componente de brillo con una palabra clave de 8 bits, donde 0 codifica el color más oscuro (es decir, el negro) y 255 el blanco. Nótese que las señales de video codificadas no necesitan comprimirse en todas las situaciones (aunque frecuentemente se hace). En caso de que lo sean, usaremos el término compresión (p. ej., mediante MPEG-HEVC, AV1, etc.), que no debe confundirse con el acto de comprimir colores en una gama o rango menor. Con brillo nos referimos a la parte de la definición del color que impactará a los colores que (se muestran) se mostrarán, la propiedad visual de ser más oscuros o más brillantes. A la luz de las tecnologías actuales, es importante comprender correctamente que puede haber dos tipos de brillo: brillo

relativo (como un porcentaje de algo, que puede no estar definido hasta que se haga una elección, p. ej., cuando el consumidor compra una determinada pantalla, y ajusta su brillo, p. ej., al 120 %, lo que hará que, p. ej., la retroiluminación emita una cierta cantidad de luz, y por lo tanto también los píxeles blancos y de color), y por otro lado, brillo absoluto. Esta última puede caracterizarse mediante la cantidad física universal: luminancia (que técnicamente se mide en la unidad nit, que también es candela por metro cuadrado). De manera que, en este texto seguiremos el enfoque de que si el brillo puede ser relativo (p. ej., porque esa subenseñanza en particular es aplicable a ambas posibilidades), lo llamaremos brillo, y si debe ser una cantidad absoluta con un valor en nit, particularmente usaremos el nombre técnico específico de luminancia. La luminosidad puede definirse como la cantidad de fotones que salen de un parche de un objeto, tal como un píxel en la pantalla, hacia el ojo (y se relaciona con el concepto de iluminación en iluminancia, ya que tal parche recibirá una determinada iluminancia, y enviará una fracción de ella hacia el espectador). También está la sensación que tendrá el ojo humano dependiendo de la cantidad de fotones entrantes, pero eso puede depender de parámetros biológicos como el ancho de la pupila o el estado químico de las células cono del ojo, etc., y esa es una discusión que usualmente no es relevante para las patentes de procesamiento de video (incluso en la

habitación del consumidor final, la pantalla puede tener como propósito mostrar algo a un espectador, pero aún puede caracterizarse simplemente por algunos fotones físicos que salen de la pantalla, haya alguien mirando o no).

5 Recientemente han surgido dos tecnologías de definición de imagen/video sin relación entre sí, que solo se unieron porque se argumentó que sería conveniente mejorar de una sola vez la calidad visual de las imágenes en todos los aspectos, pero esas dos tecnologías tienen
10 aspectos técnicos bastante diferentes.

 Por un lado, existía un esfuerzo por desarrollar una tecnología de imagen de amplia gama. Puede demostrarse que solo pueden crearse colores que se encuentren dentro del triángulo formado por los colores primarios rojo, verde y
15 azul, y ninguno fuera de él (porque las combinaciones aditivas de colores caen sobre una línea que conecta las cromaticidades mezcladas). Sin embargo, se eligieron colores primarios (en un principio, fósforos para el CRT, después, filtros de color en las pantallas LCD, etc.) que se situaban
20 relativamente cerca del lugar espectral de todos los colores existentes, de manera que se pudieran obtener colores suficientemente saturados (la saturación especifica la distancia a la que se encuentra un color respecto a los colores acromáticos, es decir, la cantidad de “color” que
25 tiene). Sin embargo, recientemente se quiso poder usar pantallas novedosas con primarios aún más saturados (p. ej.,

DCI_P3 o Rec. 2020), por lo que también es necesario poder representar los colores en espacios de color más amplios (un espacio de color es el espacio matemático 3D para representar los colores (como posiciones geométricas de números de coordenadas)). Frecuentemente se define mediante una base 2D que abarca los 3 colores primarios aditivos elegidos, definiendo cada color mediante la suma de propiedades seleccionadas de esos 3 colores primarios. A veces, en otros formalismos matemáticos, esa base triangular se representa como un círculo de color o un círculo de tonos. Para la discusión técnica, quizás sea mejor usar la palabra gama de colores, que es el conjunto de todos los colores que pueden definirse o mostrarse técnicamente (es decir, un espacio puede ser, p. ej., un sistema de coordenadas 3D que se extiende hasta el infinito, mientras que la gama puede ser un cubo de tamaño limitado en ese espacio).

En lo que respecta exclusivamente al brillo, hablaremos del rango de brillo o luminancia (más conocido como “rango dinámico”, que abarca desde un nivel de brillo mínimo hasta uno máximo). Las tecnologías actuales no se centrarán principalmente en aspectos cromáticos (es decir, el color en sí, más específicamente su saturación), sino más bien en aspectos de brillo, por lo que los aspectos cromáticos solo se mencionarán en la medida necesaria para las modalidades relevantes.

Por lo tanto, la tecnología nueva más importante de las dos direcciones de mejora es el **alto rango dinámico (HDR, por sus siglas en inglés)**. Esto no debe interpretarse como “exactamente así de alto” (ya que puede haber muchas
5 variantes de representaciones HDR, con un rango máximo sucesivamente mayor), sino más bien como “más alto que la representación de referencia/antigua: SDR” (es decir, con mayor capacidad para producir imágenes que no pueden producirse en el SDR). Dado que se necesitan nuevos
10 conceptos de codificación, también puede diferenciarse HDR de SDR por aspectos relacionados con los detalles técnicos de la representación, p. ej., la señal de video.

Una diferencia de los sistemas HDR absolutos es que definen una luminosidad única para cada píxel de la
15 imagen (p. ej., un píxel en un objeto de la imagen que sea un perro blanco al sol puede ser de 550 nits), mientras que las señales SDR solo tenían definiciones de brillo relativas (por lo que el objeto se vería, p. ej., a 75 nits (correspondiente al 94 %) en el monitor del computador de
20 alguien que podría mostrar un máximo de 80 nits, pero se mostraría a 234 nits en una pantalla SDR de 250 nits (sin embargo, el espectador típicamente no vería ninguna diferencia en el aspecto de la imagen, a menos que tuviera esas pantallas una al lado de la otra). El lector no debe
25 confundir las luminancias tal como existen (en última instancia) en la parte frontal de cualquier pantalla, con

las luminancias tal como se definen (es decir, pueden establecerse) en una señal de imagen en sí misma, es decir, incluso cuando esta se almacena pero no se muestra. Esto debe contrastarse con LDR, y la diferencia puede apreciarse incluso a nivel topológico. En SDR, la gama de colores (p. ej., representada como la tienda de campaña en la Figura 1B) es limitada (o cerrada), y termina en su extremo más brillante con un blanco del 100 %. Tales sistemas estaban diseñados para definir el conjunto de todos los colores posibles en sí mismos bajo una iluminación relativamente uniforme (la señal SDR con mejor control de calidad se obtiene en la producción de estudio, donde hay un conjunto de luces en el techo que produce una iluminación base relativamente uniforme). A veces, esta gama SDR también está (“matemáticamente deformada”) representando el blanco en el polo norte y el negro (típicamente aproximadamente 0,1 % del blanco) en el polo sur, con todos los grises en la línea que los conecta, y abarcando alrededor de ese eje de grises se encuentran todos los colores más o menos saturados de todos los tonos. La gama de colores HDR (al menos en el mundo real) es potencialmente (topológicamente) un cilindro abierto infinito, cuyo eje representa luminancias hasta el infinito, aunque en la práctica las gamas tanto de las imágenes como, por supuesto, de las pantallas físicas, terminarán en una luminancia máxima. Y en tal caso, esas gamas HDR pueden representarse con forma

de tienda de campaña, como se muestra en la Figura 1B. Otras diferencias entre las imágenes o videos HDR y los SDR son típicamente los **metadatos** compartidos que puede tener cualquier tipo de señal HDR, pero no la señal SDR.

5 Colorimétricamente, las imágenes HDR pueden representar colores más brillantes que las imágenes SDR, en particular colores más brillantes que el blanco (blancos brillantes, p. ej.). O en otras palabras, el rango dinámico será mayor. La señal SDR puede representar (para visualizar)

10 un rango dinámico de 1000:1 (la cantidad de rango dinámico que se ve realmente al visualizarlo dependerá, entre otras cosas, de la cantidad de luz ambiental que se refleja en la parte frontal de la pantalla). Entonces, si uno quiere representar, p. ej., 10.000:1 (o incluso más), debe recurrir

15 a la creación de una nueva definición de formato de imagen HDR (en general podemos usar la palabra señal si la representación de la imagen se está comunicando o se va a comunicar en lugar de, p. ejemplo, simplemente existir en el IC creador, y en general la señalización también tendrá su

20 propio formato y empaquetado, y puede emplear técnicas adicionales dependiendo del mecanismo de comunicación, tal como la modulación).

Dependiendo de la situación, el ojo humano puede ver fácilmente (incluso si todo está en una pantalla frente

25 a la cual corresponde a un pequeño ángulo de reflejo) 100.000:1 (p. ej., 10.000 nits máximo y 0,1 mínimo, que es

un buen negro para ver televisión en casa, es decir, en una habitación con iluminación tenue que solo admite una iluminación de nivel relativamente bajo, tal como por la noche). Sin embargo, no es necesario que todas las imágenes creadas por un creativo tengan un brillo tan alto: (s)puede 5 optar por que el píxel más brillante de una imagen o video tenga, p. ej., 1000 nits. Uno puede reservar la explosión en la película para el gran final, y el espectador puede querer ver una imagen diferente y más brillante allí.

10 La luminancia del blanco SDR para videos (también conocida como luminancia del punto blanco SDR (WP, por sus siglas en inglés) o luminancia máxima (ML, por sus siglas en inglés)) está estandarizada en 100 nits (que no debe confundirse con la luminancia de referencia del texto 15 blanco en imágenes HDR de 1000 nits, que es de 200 nits; esta última es el nivel de mapeo del blanco básico (objeto reflectante al 100 %) que podría provenir de una imagen SDR separada, p. ej., para ser mezclada con una imagen HDR de 1000 nits). Es decir, una representación de imagen HDR ML 20 de 1000 nits puede crear colores de objetos hasta 10 veces más brillantes (brillantes). Con esto pueden crearse, p. ej., reflejos especulares en metales, tal como el borde de un marco de ventana metálico: en SDR, la luminancia debe terminar en 100 nits, lo que hace que visualmente sean 25 ligeramente más brillantes que, p. ej., los colores gris claro de 70 nits de la parte del marco de la ventana que

no refleja especularmente. En HDR pueden hacerse que los píxeles que reflejan la fuente de luz hacia el ojo, p. ej., a 900 nits, brillen de manera agradable, dando a la imagen un aspecto naturalista, como si fuera una escena real. Lo mismo puede hacerse con bolas de fuego, bombillas, etc.

Esto se refiere a la definición (codificación) de las imágenes. Como una pantalla que solo puede mostrar blancos con un brillo de hasta 650 nits (la luminancia máxima de la pantalla ML_D) va a mostrar realmente las imágenes es un asunto completamente diferente, concretamente una cuestión de **adaptación de la pantalla, también conocida como ajuste de la pantalla**, no de (de)codificación de imágenes. Asimismo, la relación con la manera en que una cámara captura los colores de una escena HDR puede ser estrecha o más flexible: en general, asumiremos que los colores HDR ya se han definido en la imagen HDR cuando hablemos de tecnologías tales como la codificación, la comunicación, la conversión de rango dinámico y lo similar. De hecho, los colores originales capturados por la cámara, o más específicamente su luminancia, pueden haber sido cambiados a valores diferentes, p. ej., por un graduador de color humano (que define el aspecto final de una imagen, es decir, qué valores de triplete de color debe tener cada color de píxel de la(s) imagen(es)), o por algún algoritmo automático. Entonces, teniendo en cuenta que el tamaño de la gama cromática puede

ampliarse en menos de un factor de 2, mientras que el rango de brillo, p. ej., el rango de luminosidad, puede ampliarse en un factor de 100, cabe esperar diferentes fundamentos técnicos y soluciones para las dos tecnologías de mejora.

5 Una imagen HDR puede estar asociada con un metadato llamado luminosidad del punto blanco de la pantalla maestra (MDWPL, por sus siglas en inglés), también conocido como ML_V. Este valor, que típicamente se comunica en los metadatos de la señal, y es un caracterizador de las imágenes
10 de video HDR (en lugar de una pantalla específica, ya que es un elemento de una pantalla virtual asociada específicamente con el video, siendo una pantalla ideal prevista para la que se han optimizado los colores de los píxeles del video para que se ajusten). Este es un parámetro configurable del video,
15 que puede considerarse similar a la elección de la relación de aspecto del lienzo de pintura por parte de un pintor: primero, el pintor elige una AR adecuada, p. ej., 4:1 para pintar un paisaje, o 1:1 cuando quiere hacer una naturaleza muerta, y después de eso comienza a colocar de manera óptima
20 todos los objetos en el lienzo de pintura elegido. En una imagen HDR, el creador, después de haber establecido que el MDWPL es, p. ej., de 5000 nits, realizará sus elecciones secundarias para que, en una escena específica, la pantalla de la lámpara tenga 700 nits, las llamas de la chimenea se
25 distribuyan alrededor de 500 nits, etc.

El aspecto visual primario de las imágenes HDR es (dado que los negros son más complejos) los colores brillantes adicionales (de manera que pueden definirse rangos con solo un valor MDWPL, si se supone que la
5 luminosidad inferior está fijada, p. ej., a 0,1 nit). Sin embargo, la creación de imágenes HDR también puede involucrar negros más profundos, hasta, p. ej., 0,0001 nit (aunque eso es principalmente relevante para entornos de visualización oscuros, tales como en salas de cine).

10 Los otros objetos, p. ej., los que simplemente reflejan la luz de la escena, se coordinarán para que sean, p. ej., al menos 40 veces más oscuros en un video con un nivel de luminosidad MDWPL de 5000 nits, y p. ej. al menos 20 veces más oscuros en un video de 2000 nits, etc. Por lo
15 tanto, la distribución de las luminancias de todos los píxeles de la imagen dependerá típicamente del valor MDWPL (lo que no hará que la mayoría de los píxeles sean muy brillantes).

La codificación digital del brillo implica una
20 magnitud técnica llamada luma (Y). A las luminancias las representaremos con la letra L , y a los brillos (relativos) con la letra B . Nótese que técnicamente, p. ej., para facilitar la definición de algunas operaciones, siempre pueden normalizarse las luminancias del rango WPDPL de
25 5000 nits al rango normalizado $[0,1]$, pero eso no quita que estas luminancias normalizadas sigan representando

luminancias absolutas en un rango que termina en 5000 nits
(en contraste con los brillos relativos, que nunca tuvieron
un valor de luminancia absoluta claramente asociado y solo
pueden convertirse a luminancias ad hoc, típicamente con
5 algún valor arbitrario).

Para las señales SDR, la codificación de luma
usaba una denominada función de transferencia opto-
electrónica (OETF, por sus siglas en inglés), entre los
brillos ópticos y los códigos de luma electrónicos,
10 típicamente de 8 bits, que por definición era:

$$Y_float = \sqrt{B_relative}.$$

Si $B_relative$ es un número flotante que varía de 0 a 1,0,
15 Y_float también lo será.

Posteriormente, ese valor de señal Y_float se
cuantifica, porque queremos representaciones digitales de 8
bits, por lo tanto, el valor Y_dig que se comunica a los
receptores a través de, p. ej., vías aéreas DVB (o video a
20 pedido suministrado por Internet, o disco Blu-ray, etc.)
tiene un valor entre 0 y 255 (es decir, $potencia(2;8)-1$).

Puede mostrarse la gama de todos los colores SDR
(o similarmente pueden mostrarse las gamas de colores HDR,
que si se usan los mismos colores primarios RGB que definen
25 la gama cromática tendrían la misma base, pero después de
la renormalización se estiran verticalmente a una gama de

colores absoluta más amplia) como en la **Figura 1B**. Los valores de Cb y Cr más grandes darán lugar a una saturación (sat) mayor (caracterización del color psicovisualmente más relevante), que se desplaza hacia afuera desde el eje vertical de colores insaturados o incoloros que muestra los colores acromáticos en el medio, hacia los colores máximamente saturados en el círculo (que es una transformación del triángulo de color usual abarcado por los colores primarios RGB como vértices). Los tonos h (es decir, la categoría de color, amarillos, vs. verdes, versus azules) serán ángulos a lo largo del círculo. El eje vertical representa luminancias normalizadas (en una representación de gama de colores lineal), o lumas normalizadas (en una representación no lineal (codificación) de esas luminancias, p. ej., mediante una OETF psicovisualmente uniformizada o su inversa la EOTF). Dado que después de la normalización (es decir, la división por los respectivos valores MDWPL, p. ej., 2000 para una imagen HDR de un video en particular y 100 para una imagen SDR), la representación común será sencilla y se podrán definir funciones de mapeo de luminancia (o luma) en ejes normalizados como se muestra en la **Figura 1D** (es decir, la función F_{comp} redistribuye los distintos valores de píxeles del objeto de la imagen según sea necesario, de manera que en la representación de luminancia real, p. ej., un objeto oscuro parezca igual, es decir, tenga la

misma luminancia, pero una luminancia normalizada diferente, ya que en una situación la luminancia normalizada de un píxel se multiplicará por 100 y en la otra por 2000, por lo que para obtener la misma luminancia final, el último píxel debería tener una luminancia normalizada de 1/20 de la del primero). Al reducir el rango de luminancias, normalmente se obtiene (en la representación normalizada a 1,0, es decir, mapeando un rango de luminancias de entrada normalizadas L_{in} entre
5 cero y uno a luminancias de salida normalizadas L_{out}) una función convexa que se encuentra en todas partes por encima de la diagonal diag, pero la forma exacta de esa función F_{comp} , p. ej., cómo tiene que aumentar en los negros, dependerá típicamente no solo de los dos valores MDWPL,
10 sino también (para tener la versión más perfecta de la tecnología de regradación) del contenido de la escena de varias imágenes (de video o fijas), p. ej., si la escena es una cueva oscura, y hay acción ocurriendo en un área sombreada, que aún debe ser razonablemente visible incluso
15 en rangos de luminancia de 100 nits (de ahí el fuerte aumento de los negros para tal escenario, en comparación con una escena de luz diurna que puede emplear una función casi lineal que casi se superpone con la diagonal).

En la representación de la Figura 1B puede
25 mostrarse el mapeo de un color HDR (C_H) a un color LDR (C_L) de un píxel como un desplazamiento vertical

(suponiendo que ambos colores deben tener el mismo color propio, es decir, tono y saturación, que usualmente es el requisito técnico deseado, es decir, en el plano circular del suelo se proyectarán al mismo punto). Y e significa el color amarillo, y su color complementario en el lado opuesto del eje acromático de luminancias (o lumas) es el azul (B), y W significa blanco (el color más brillante en la gama de colores, también conocido como el punto blanco de la gama, con los colores más oscuros, los negros, en la parte inferior). Los brillos (que son proyecciones de colores sobre el eje acromático de grises que comienza en el negro más profundo, que frecuentemente se asume simplemente como cero, y termina en el brillo máximo), tal como primero, el brillo LDR Br_L y segundo, el brillo HDR Br_H , pueden ser adimensionales (típicamente escalados a un máximo de 1,0), pero normalmente están asociados con algunas luminancias que se van a mostrar. En las últimas representaciones de imágenes HDR, cada píxel define una luminancia real para visualizar; es decir, típicamente se supone que cualquier pantalla con capacidad de visualización de luminancia máxima, capaz de mostrar los píxeles de imagen más brillantes, debe seguir la memoria descriptiva de color de la imagen codificada y mostrar cada píxel (al menos aproximadamente) con la luminancia que se le ha asignado en la imagen. Por supuesto, cualquier pantalla puede optar por aclarar (u oscurecer) la imagen ligeramente por su

cuenta, pero si muestra la imagen exactamente como la ajustó el creador, mostrará las luminancias definidas en la imagen. Por lo tanto, el conocimiento de las luminancias de los píxeles ya está en la imagen, independientemente de si se mostrará o cómo se mostrará realmente (esto puede formularse asociando la imagen con una pantalla virtual en lugar de una real, p. ej., diciendo que la imagen es óptima para una pantalla de 4000 nits, porque un píxel de 4000 nits en la imagen tiene un brillo óptimo para mostrarlo en una pantalla real con una capacidad máxima de 4000 nits).

En la Figura 1B, aunque hemos mostrado una representación de gama de colores simple donde todos los colores HDR y todos los colores LDR tienen brillos normalizados entre 0 y 1,0, típicamente también se asocian algunas luminancias absolutas a esos colores, p. ej., asociando una luminancia del punto blanco de la pantalla maestra (MDWPL). Cualquier luminancia absoluta se encuentra multiplicando la luminancia normalizada por el valor MDWPL. Entonces, p. ej., los colores definidos en una escala absoluta de luminancias de un cuantificador perceptual (p. ej., codificados en YCbCr) pueden asociarse con un MDWPL de, p. ej., 5000 nit, lo que significa que la(s) imagen(es) creada(s) es (son) perfecta(s) (sin necesidad de mapeo de luminancia en una pantalla receptora, ni hacia abajo ni hacia arriba, en principio) para una pantalla con una luminancia máxima real de 5000 nits. Esto se refiere a la

definición de una imagen (per se), que típicamente es algo más que la visualización real: cualquier pantalla receptora podría desear cambiar la imagen antes de mostrarla, p. ej., hacerla un 10 % más brillante. Un brillo de 0,5 o 50 % puede entonces, si se cuantifica linealmente, indicar que tal píxel debe mostrarse a 2500 nits. El UHD Alliance ha indicado que 1000 nits MDWPL es un buen punto de partida para obtener HDR (con calidad mínima) en una pantalla (lo cual puede estar indicado en la pantalla por su logotipo UltraHD Premium). Como este nivel de brillo máximo es 10 veces más brillante que un nivel LDR típico de (aproximadamente/alrededor de) 100 nits (que se convirtió en el nivel estándar para definir los videos LDR en la era HDR), pueden usarse los 100-1000 nits adicionales o hasta 10 veces más brillante que el blanco de un objeto que refleja difusamente (un ejemplo de un blanco lambertiano difuso sería un trozo de papel o una camisa blanca con la iluminación promedio base de la escena) para efectos HDR tal como una iluminación extra fuerte de algunas regiones de algunos objetos, o aproximaciones de reflejos especulares, o explosiones, u otros objetos autoluminosos, etc. Por supuesto, puede ser mejor si en una imagen HDR de mayor calidad con un valor MDWPL asociado más alto pueden, p. ej., hacerse que algunos de los píxeles más brillantes tengan, p. ej., 2000 nits. P. ej., puede quererse que el sol brille 50 veces más que 100 nits, y tal imagen MDWPL

de 5000 nits podría dar un aspecto aún mejor para la escena HDR que una gradación MDWPL de 2000 nits de esa misma escena, pero incluso 1000 nits MDWPL ya ofrecen bastante margen para obtener buenos efectos HDR e imágenes HDR impresionantes. Pero todo el marco desarrollado alrededor de la codificación HDR se ha diseñado de tal manera que el creador de video o imagen no tenga que adivinar cuáles serán las capacidades típicas de las pantallas en el futuro, y pueda crear imágenes de acuerdo con sus deseos (y las pantallas se encargarán de la discrepancia, típicamente basándose en recetas también definidas por el creador sobre la mejor manera de reajustar el color de sus imágenes).

Si se codifica una imagen en un formato independiente de los nits (o relativo, o totalmente independiente de la pantalla), tal como Hybrid Loggamma, puede predefinirse, aunque en teoría esas lumas de píxeles que codifican el brillo no tienen luminancias asociadas, (adicionalmente) que la luma HLG más brillante corresponde a, por ejemplo, 1000 nits. En ese caso, también esos píxeles podrían asociarse con sus respectivas luminancias correspondientes. Los defensores de HLG pueden insistir en que los videos HDR codificados con HLG no deberían tener luminancias específicas asociadas a los píxeles, pero si es necesario mezclarlos con otro contenido de luminancia absoluta, tal asignación de una luminancia HLG máxima óptima lo hace posible. P. ej., un aparato receptor podría

asignar un valor de luminancia razonable al píxel más brillante codificable en HLG, p. ej., porque se estima que se coordina bien con alguna otra definición de imagen.

Nótese que en una representación normalizada (es decir, el valor máximo es, p. ej., típicamente 1,0, al dividir el valor del píxel más brillante posible, digamos 5000 nits, por su metadato comunicado, que normalmente también debería ser 5000), los brillos LDR (normalizados) típicamente son más altos que los brillos HDR, lo que puede parecer algo contradictorio con que sus valores absolutos sean típicamente más bajos que los de HDR (o al menos iguales para objetos más oscuros). De hecho, al mostrar una imagen HDR, se podría esperar que existieran algunos píxeles más brillantes en al menos alguna parte de la imagen HDR (p. ej., una explosión y el reflejo de esa luz brillante en los objetos circundantes, pero los objetos en las zonas sombreadas no tienen por qué ser más brillantes) que en la imagen SDR de la misma escena. Pero si dividimos por una luminancia máxima más alta (p. ej., 5000 nits), algún objeto oscuro en la imagen HDR terminará con una luminancia relativa menor en la imagen HDR que en la imagen SDR, donde la misma luminancia absoluta se divide normalmente solo por 100.

Por lo tanto, el lado receptor (tanto en el pasado como en la actualidad) sabrá que tiene un video SDR, si recibe este formato. El blanco máximo (de SDR) será por definición el color más brillante que SDR pueda

definir. Entonces, si ahora se quieren obtener colores de imagen más brillantes (p. ej., de lámparas luminosas reales), eso debe hacerse con un códec diferente (como puede demostrarse matemáticamente, el Rec. 709 OETF solo
5 permite una codificación de hasta 1000:1 y no más).

Por lo tanto, se definen nuevos marcos con diferentes funciones de asignación de código (EOTF o OETF). Lo que interesa aquí es principalmente la definición de los códigos de luma.

10 Por razones que escapan al alcance de la presente discusión, la mayoría de los códecs HDR comienzan definiendo una función de transferencia electro-óptica en lugar de su inversa, la OETF. Entonces, al menos pueden definirse básicamente los colores más claros (y más oscuros). Eso como
15 tal no es suficiente para un sistema de codificación HDR profesional, ya que, al ser diferente del SDR e incluso tener varios tipos, se necesita más información técnica (nueva en comparación con la codificación SDR) relacionada con las imágenes HDR, que serán los metadatos.

20 La propiedad de esas EOTF HDR es que son mucho más pronunciadas, para codificar un rango mucho mayor de luminancias HDR que necesitan ser codificadas, y una parte significativa de ese rango codifica específicamente colores más oscuros (relativamente más oscuros, ya que
25 aunque uno puede estar codificando luminancias absolutas con, p. ej., la EOTF del cuantificador perceptual (PQ, por

sus siglas en inglés) (estandarizada en SMPTE 2084), uno aplica la función después de la normalización). De hecho, si se usaran funciones de potencia exactas como las EOTF para codificar las luminancias HDR como lumas HDR, se obtendría una potencia de 4, o incluso de 7. Cuando un receptor recibe una señal de imagen de video definida por una EOTF (p. ej., un cuantificador perceptual), sabrá que se trata de un video HDR. Será necesario que la EOTF pueda decodificar las lumas de los píxeles en el plano de lumas que abarca la imagen (es decir, con un ancho de, p. ej., 4000 píxeles y una altura de 2000), que simplemente serán números binarios. Típicamente, las imágenes HDR también tendrán una longitud de palabra más grande, p. ej., 10 bits. Sin embargo, no se debe confundir la codificación no lineal que puede diseñarse a voluntad optimizando una forma EOTF no lineal con las codificaciones lineales y la cantidad de bits necesarios para ellas. Si se necesita controlar, con un código lineal (representado por bits), p. ej. un píxel DMD, para alcanzar, p. ej., una modulación de 10000:1 de oscuro a brillante, es necesario tomar el $\log_2$ para obtener el número de bits. Allí se necesitarían al menos 14 bits (que por razones técnicas pueden redondearse a 16 bits), ya que $\text{power}(2;14) = 16384 > 10000$. Pero al poder diseñar de manera inteligente la forma de la EOTF, y sabiendo que el sistema visual no percibe todas las diferencias de luminancia por igual, el actual

solicitante ha demostrado que (sorprendentemente) pueden comunicarse señales de televisión HDR bastante razonables con solo 8 bits de componente de color por píxel (por supuesto, si es técnicamente factible en un sistema, 10 bits pueden ser mejores y con mayor preferencia). Por lo tanto, el lado receptor puede en ambas situaciones recibir como entrada un color de píxel codificado (luma y Cb, Cr; o en algunos sistemas mediante la transformación matricial de los valores de componentes $R'G'B'$ no lineales equivalentes) que enlaza entre 0 y 255, o 0 y 1023, pero sabrá el tipo de señal que está recibiendo (por lo tanto, lo que debería mostrarse) a partir de los metadatos, tales como los metadatos (p. ej., metadatos MPEG VUI) EOTF cocomunicados (p. ej., un valor 16 que significa PQ; 18 significa que se usó otra OETF para crear las lumas, concretamente la OETF Hybrid LogGamma, por lo que se debe usar la inversa de esa función para decodificar el plano de luma), en muchas codificaciones HDR el valor MDWPL (p. ej., 2000 nit), y en codificaciones HDR más avanzadas, metadatos adicionales (algunas pueden, p. ej., cocomunicar funciones de mapeo de luminancia o luma para aplicar el mapeo de luminancias de imagen desde un rango dinámico de luminancia primario a un rango dinámico de luminancia secundario, tal como una función FL_{enc} por imagen).

Por lo tanto, típicamente, un receptor puede comprender si tenemos una imagen SDR o una imagen HDR (o, de

hecho, cuál es la diferencia) en al menos uno de dos niveles. A nivel colorimétrico, las imágenes HDR típicamente tienen al menos algunos colores de píxeles más brillantes que el color más brillante de la imagen SDR, que típicamente termina en su extremo brillante en o alrededor del blanco de reflexión difusa (un objeto con un 100 % de reflexión, bajo la iluminación “promedio”, predominante o base). En términos de nits, puede definirse previamente este valor de nit en la señal de imagen, p. ej., típicamente 100 nits, o cualquier valor cercano a 100 nits. Las imágenes HDR contienen píxeles mucho más brillantes, al menos un par de veces, p. ej., 4 veces, más de 5 veces, 10 veces, o incluso más brillantes (que el blanco lambertiano al 100 %). Entonces, si una señal comprende un valor MDWPL de al menos 1000 nits o superior, se sabe que se trata de una imagen HDR. También puede esperarse que el creador de la imagen haya ajustado la composición de la escena de tal manera que algunos de los objetos de la imagen (los objetos difusos) tengan luminancias que oscilen, p. ej., entre 0 y 100 nits, o valores cercanos a esos, y que haya aquí y allá algunos objetos más brillantes, como la llama de una vela, o algún elemento luminoso fuera del campo de visión que se refracta hacia la cámara por el efecto de la lente de una botella de agua, etc.

Por otro lado, a nivel de señalización, el decodificador puede entender, a partir de las diversas opciones de codificación, las codificaciones de imágenes HDR

vs. las imágenes SDR (las imágenes HDR contienen al menos un aspecto adicional y/o diferente; cualquier cosa que no existiera en la codificación de imágenes SDR heredada, como p. ej. un valor MDWPL como un metadato estandarizado cocomunicado, indica que estamos tratando con una imagen o video HDR, y en particular con uno de la clase que comunica este metadato). Por lo tanto, las imágenes SDR heredadas no tienen ningún valor MDWPL asociado (típicamente cocomunicado en los metadatos de las imágenes, es decir, las matrices de componentes de color de píxeles), o de hecho no especifican ninguna luminancia específica para ninguno de los píxeles. También tienen una EOTF fija asumida y no cocomunicada (rec. 709), ya que existía “solo SDR”, que simplemente codificaba imágenes o videos de lo que había, es decir, el conjunto de colores de objetos iluminados de manera mayormente uniforme; la EOTF tampoco es capaz de alcanzar un rango o una relación de brillos o luminancias tan amplios como los que tienen típicamente las imágenes HDR; es decir, pueden lograr una relación de luminancia de más oscuro a más brillante de aproximadamente 1000:1 en lugar de, p. ej., 1000 nits: 0,1 nit es decir 10.000:1, o incluso rangos dinámicos de luminancia de más de un millón a 1. Varios estándares de comunicación de video HDR permiten no solo comunicar una gama tan amplia de luminancias, sino también elegir las EOTF HDR más adecuadas entre varias posibles. Si un receptor ve una indicación EOTF (p. ej., EOTF PQ o EOTF HLG, necesaria

para decodificar correctamente el YCbCr recibido, p. ej.,
códigos de 10 bits de las matrices de color de la imagen),
que típicamente se comunica como un metadato que es un
número que describe uno específico de un conjunto de
5 posibles EOTF HDR, p. ej., que recibe luma HLG o luma PQ,
se entiende que está recibiendo imágenes HDR novedosas en
lugar de imágenes SDR estándar antiguas. Y además se
entiende que la codificación HDR se usó para crear los
valores de luma de píxeles específicos tal como se
10 recibieron. Si bien hemos añadido lo anterior para completar
la información y presentarla correctamente a los lectores
menos experimentados, la persona experta ya debería
comprender suficientemente los detalles subyacentes de la
codificación y decodificación HDR básicas, de manera que
15 pueda pasar a los detalles innovadores de la presente
aplicación, es decir, lo que debería aprender aquí.

Detallamos las necesidades típicas de una cadena
de procesamiento de imágenes HDR ya más sofisticada con la
ayuda de una explicación sencilla **Figura 1** (para una imagen
20 típica de una escena HDR agradable, de un monstruo al que se
combate en una cueva con un lanzallamas, cuya **gradación**
maestra (Mstr_HDR) se muestra espacialmente en la Figura 1A,
y el rango de luminancias de píxeles que aparecen a la
izquierda de la Figura 1C). La gradación maestra o imagen
25 con gradación maestra es aquella en la que el creador de la
imagen puede hacer que su imagen resulte tan impresionante

(p. ej., realista) como desee. P. ej., en una película navideña, puede hacer que el escaparate de una panadería parezca algo iluminado haciendo que las paredes amarillas sean un poco más brillantes que el blanco papel, p. ej., 150 nits (y un amarillo realmente intenso en lugar de un amarillo pálido), y las bombillas pueden tener 900 nits (lo que dará a la imagen un aspecto realmente iluminado y navideño, en lugar de uno apagado en el que todas las luces se recortan en blanco y no son mucho más brillantes que el resto de los objetos de la imagen, tal como el verde del árbol de Navidad).

Por lo tanto, lo fundamental es poder codificar (y, típicamente, también decodificar y visualizar) objetos de imagen más luminosos que en una imagen SDR típica.

Veámoslo ahora colorimétricamente. El SDR (y su codificación y señalización) se diseñó para poder reproducir cualquier color de reflexión lambertiana (es decir, un objeto típico, como unos jeans azules, que absorbe parte de la luz incidente, p. ej., las longitudes de onda roja y verde para emitir únicamente luz azul al espectador o a la cámara que lo graba) bajo una iluminación uniforme y adecuada (de la escena en la que la cámara graba la acción). Igual que haríamos con un cuadro: si no añadimos pintura, obtendremos todo el brillo que refleja el lienzo de pintura blanco, y si añadimos una capa gruesa de pintura muy absorbente, veremos un trazo o un punto negro. Podemos representar todos los colores más brillantes que el negro

más intenso y más oscuros que el blanco en lo que se conoce como gama de colores representables, tal y como se muestra en la Figura 1B (la “tienda de campaña”). Como plano inferior, tenemos un círculo que abarca todas las cromaticidades representables (nótese que pueden tenerse largas discusiones sobre si en un sistema RGB típico esto debería ser un triángulo, pero esos detalles están fuera del alcance de las presentes enseñanzas). Una cromaticidad se compone de un cierto tono h (ángulo de rotación) (p. ej., verde azulado, p. ej., “turquesa”), y una saturación sat , que es la cantidad de color puro mezclado en un gris, p. ej., la distancia desde el eje vertical en el medio que representa todos los colores acromáticos desde el negro en la parte inferior volviéndose cada vez más brillantes hasta que llegamos al blanco. Los colores cromáticos, p. ej., un púrpura semisaturado, también pueden tener brillo, el mismo color puede ser un poco más oscuro o más brillante. Sin embargo, el color más brillante en un sistema aditivo solo puede ser el blanco (incoloro), ya que se obtiene ajustando todos los canales de color al máximo ($R=G=B=255$); por lo tanto, no hay desequilibrio que haga que el color sea claramente rojo (todavía existe un poco de azulado o amarillento en la cromaticidad del punto blanco elegido, pero esa es una discusión adicional innecesaria, asumiremos el blanco luz diurna D65). Podemos definir esos colores SDR estableciendo MDWPL en un 100 % (relativo) para el blanco W

(nota: en SDR, el blanco en realidad no tiene una luminancia, ya que el SDR antiguo no tiene una luminancia asociada a la imagen, pero podemos fingir que es X nit, p. ej., típicamente 100 nits, que es un buen valor representativo promedio de los distintos televisores SDR antiguos).

Ahora queremos representar colores más brillantes que los lambertianos, p. ej., el objeto de llama autoluminiscente (flm) del lanzallamas del soldado (sol) que lucha contra el monstruo (mon) en esta cueva oscura.

10 Digamos que definimos una gradación HDR maestra de 5000 nits (luminancia máxima de video ML_V) (maestra significa la imagen inicial, la más importante en este caso, la gradación de mejor calidad, que graduaremos de manera óptima primero, para definir el aspecto de esta imagen de escena HDR, y de la cual podemos derivar gradaciones secundarias, también conocidas como imágenes graduadas, según sea necesario). Para simplificar, hablaremos de lo que sucede con las luminancias (universales), y por ahora podemos dejar fuera de la discusión el debate sobre los

15 códigos de luma correspondientes. De hecho, PQ puede codificar entre 1/10.000 nits y 10.000 nits, por lo que no hay problema en comunicar esas luminancias de píxeles graduadas como, p. ej., una imagen HDR pixelada YCbCr de 10 bits por componente, si se codifica de acuerdo con esa

20 EOTF PQ (por supuesto, los mapeos también pueden

representarse, y p. ej. implementar en las unidades de IC de procesamiento, como un mapeo de luma equivalente).

Las dos líneas horizontales punteadas representan las limitaciones de la imagen codificable SDR, al asociar 100 nits con el 100 % del blanco SDR.

Aunque en una cueva el monstruo estará fuertemente iluminado por la luz de las llamas, le daremos una luminancia (promedio) de 300 nits (con alguna variación, debido a la Ley del cuadrado inverso de la atenuación de la luz, la textura de la piel, etc.).

El soldado puede tener 20 nits, ya que es un valor ligeramente oscuro que aún proporciona una buena visibilidad básica.

Un vehículo (veh) puede estar oculto en algún rincón sombrío y, por lo tanto, en una escena HDR de buen impacto arquetípica de una cueva, p. ej., tener una luminancia de 0,01 nit. Las llamas pueden hacerse impresionantemente brillantes (aunque sin exagerar). En un rango HDR disponible de 5000 nits, podríamos elegir 2500 nits, alrededor del cual aún podríamos crear gradualmente algunas partes más oscuras y más brillantes, pero todas con colores agradables (amarillo y tal vez algunos naranjas).

¿Qué pasaría ahora en una representación SDR típica, p. ej., al capturar una imagen SDR directamente desde la cámara?

El operador de cámara abriría el diafragma para que el soldado apareciera a “20 nits”, o más precisamente, al 20 %. Dado que las llamas son mucho más brillantes (nota: en realidad no mostramos las luminancias de la escena del mundo real, ya que el video HDR maestro Mstr_HDR ya es una gradación óptima para tener el mejor impacto en un escenario de visualización típico de una sala de estar, pero también en el mundo real las llamas serían bastante más brillantes que el soldado, y ciertamente el vehículo), todas se recortarían al blanco máximo. Entonces, veríamos una zona brillante, sin detalles, y tampoco amarilla, ya que el amarillo debe tener una luminancia menor (por supuesto, el director de fotografía podría optimizar la imagen para que todavía se vea algo de llama incluso en LDR, pero, en primer lugar, eso nunca resultaría tan impactante por su mayor brillo y, en segundo lugar, iría en detrimento de los demás objetos, que tendrían que quedar más oscuros).

Lo mismo ocurriría si fabricáramos un televisor SDR (máx. 100 nits) que representara la equiluminancia, es decir, que representara con precisión todas las luminancias de la gradación HDR original que pudiera reproducir, pero recortara todos los objetos más brillantes a un blanco de 100 nits.

Por lo tanto, el paradigma habitual en la era LDR consistía en asignar de manera relativa, es decir, el brillo máximo (en este caso, la luminancia) de la imagen recibida

a la capacidad máxima de la pantalla. Para estos mapeos de 5000 nits por división por 50 en 100 nits, las llamas seguirían viéndose bien, ya que se distribuyen como amarillos y naranjas de aproximadamente 50 nits (que es un
5 brillo representable para un amarillo, ya que, como vemos en la Figura 1B, la tienda de campaña de gama de colores para los amarillos solo desciende ligeramente en luminancia al acercarse a los amarillos más saturados, en contraste con los azules (B) en el otro lado del segmento para este
10 ángulo de tono B-Ye, que solo pueden reproducirse en versiones relativamente oscuras). Sin embargo, esto iría en detrimento de que todo lo demás se oscureciera bastante, p. ej., el soldado de 20/50 nits, que es completamente negro (y este es un problema típico que vemos en los renderizados
15 SDR de este tipo de escenas de películas).

Entonces, si se ha establecido una buena luminancia máxima HDR (es decir, ML_V) para la gradación maestra, y una buena EOTF, p. ej., PQ para su codificación, en principio podemos empezar a comunicar imágenes HDR a
20 receptores, p. ej., pantallas de televisión para el consumidor, computadores, proyectores de cine, etc.

Pero este es solo el sistema HDR más básico.

El problema es que, a menos que el lado receptor tenga una pantalla capaz de mostrar píxeles con un brillo
25 mínimo de 5000 nits, sigue sin estar claro cómo se van a mostrar esos píxeles.

Es necesario realizar un mapeo descendente de luminancia (adaptación DR) en el televisor para que los píxeles más oscuros se puedan visualizar. P. ej., si la pantalla tiene una **luminancia máxima de pantalla ML_D**
5 (usuario final) de 1500 nits, de alguna manera se podría intentar calcular 1200 nits de píxeles amarillos para la llama (potencialmente con errores, como alguna decoloración, p. ej., cambiar los naranjas a amarillos).

Este mapeo descendente de luminancia no es
10 realmente una tarea fácil, especialmente hacerlo con mucha precisión en lugar de suficientemente bien, y por lo tanto se han inventado diversas tecnologías (también para la tarea no necesariamente similar de mapeo ascendente de luminancia, para crear una imagen de salida de mayor rango
15 dinámico y en particular, con una luminancia máxima superior a la de la imagen de entrada).

Típicamente, se busca una función de mapeo (genéricamente, es decir, usada para simplificar la explicación) de forma convexa en un gráfico de luminancia (o
20 brillo) normalizada, como se muestra en la Figura 1D. Tanto la luminancia de entrada como la de salida se definen aquí en un rango normalizado cuyo máximo es igual a uno, pero se debe tener en cuenta que, en el eje de entrada, este valor corresponde, p. ej., a 5000 nits, y en el eje de salida, p.
25 ej., a 200 nits (lo cual puede implementarse fácilmente mediante división o multiplicación, según el caso). En una

representación normalizada de este tipo, los colores más oscuros son típicamente demasiado oscuros para la gradación con el rango dinámico más bajo de las dos imágenes (en este caso, para la conversión a baja resolución, mostrada en el

5 eje vertical de salida, de luminancias de salida normalizadas L_{out} , mientras que el eje horizontal muestra todas las luminancias de entrada normalizadas posibles L_{in}). Por lo tanto, para tener una imagen de salida satisfactoria que corresponda a la imagen de entrada, debemos aumentar

10 relativamente esas luminancias más oscuras, p. ej., multiplicándolas por $3x$, que es la pendiente de esta función de compresión de luminancia F_{comp} para su extremo más oscuro. Pero no puede aumentarse indefinidamente si no se quiere que ningún color se recorte al máximo de salida; por

15 lo tanto, la curva debe tener una pendiente cada vez menor para las luminancias de entrada más brillantes; p. ej., típicamente puede asignar una entrada de 1,0 a una salida de 1,0. En cualquier caso, la función de compresión de luminancia F_{comp} para la degradación se situará normalmente

20 por encima de la diagonal de 45 grados (diag).

Es necesario tener cuidado para hacerlo correctamente. P. ej., a algunas personas les gusta aplicar tres funciones de compresión de este tipo a los tres canales: rojo, verde y azul, por separado. Aunque se trata de una

25 garantía sencilla y buena de que todos los colores encajarán en la gama de salida (un cubo RGB, que en la representación

de cromaticidad-luminancia (L) se convierte en la tienda de
campana de la Figura 1B), especialmente en casos de mayor
no linealidad, puede dar lugar a errores de color
significativos. Un tono naranja rojizo, p. ej., se determina
5 por el porcentaje de rojo y verde, p. ej., 30 % verde y 70 %
rojo. Si ahora la función de mapeo duplica ese 30 %, pero
el rojo permanece casi sin cambios en la parte de pendiente
suave de la función de mapeo, obtendremos un 60/70, es decir,
un 50/50, es decir, un amarillo en lugar de un naranja. Esto
10 puede resultar especialmente molesto si depende (a
diferencia del paradigma SDR) de una iluminación no uniforme
de la escena; p. ej., un auto deportivo que entra en la
sombra y de repente se vuelve amarillo.

Por lo tanto, aunque la forma general deseada para
15 iluminar los colores siga siendo la función F_{comp} (p. ej.,
determinada por el creador del video al realizar la gradación
de una imagen secundaria correspondiente a su imagen HDR
maestra, que ya ha sido óptimamente graduada), se busca un
mapeo descendente más preciso. Como se muestra en la Figura
20 1B, en muchos escenarios puede desearse realizar una
regradación que simplemente cambie el brillo del componente
de luminancia normalizada (L), pero sin alterar el tipo de
color intrínseco, es decir, su cromaticidad (tono y
saturación). Si tanto SDR como HDR se representan con los
25 mismos colores primarios: rojo, verde y azul, tendrán una
tienda de campana de gama de colores de manera similar, solo

que una tendrá una representación de luminancia absoluta mayor que la otra. Si se escalan ambas gamas con sus respectivos valores MDWPL (p. ej., MDWPL1 = 100 nits y MDWPL2 = 5000 nits), ambas gamas se superpondrán exactamente. El mapeo deseado de un color HDR C_H a un color SDR de salida correspondiente C_L (o viceversa) será simplemente un desplazamiento vertical, mientras que la proyección al círculo del plano de cromaticidad permanece igual.

Aunque los detalles de tales enfoques también están fuera del alcance de la presente aplicación, ya hemos considerado ejemplos de tal mecanismo de mapeo de color, donde los tres componentes de color se procesan coordinadamente, aunque en una ruta de procesamiento de luminancia y croma separada, p. ej., en WO2017157977.

Si ahora es posible realizar una degradación con una (o más) funciones de mapeo de luminancia (la forma puede ser optimizada por el creador del (de los) video(s)), al usar funciones invertibles puede diseñarse un códec HDR más avanzado.

En lugar de simplemente hacer alguna gradación secundaria final a partir de la imagen maestra, p. ej., en un televisor, puede crearse una versión de imagen de rango dinámico más bajo para la comunicación, **imagen de comunicación IM_Comm**. En este ejemplo, hemos elegido que esta imagen se defina con una luminancia máxima de imagen de comunicación ML_C igual a 200 nits. La imagen original

de 5000 nits puede ser reconstruida (también conocida como decodificada) como una imagen reconstruida Rec_HDR (es decir, con la misma luminancia máxima de imagen reconstruida ML_REC) por los receptores, si reciben en los metadatos la

5 función de mapeo de luminancia de decodificación FL_dec, que típicamente es sustancialmente la inversa de la función de mapeo de luminancia de codificación FL_enc, que fue usada por el codificador para mapear todas las luminancias de píxeles de la imagen HDR maestra a las correspondientes

10 luminancias de píxeles más bajas de la imagen de comunicación Im_comm. Por lo tanto, la imagen proxy para comunicar una imagen con un rango dinámico más alto (DR_H, p. ej., que abarca desde 0,001 nit hasta 5000 nits) es una imagen con un rango dinámico diferente y más bajo (DR_L).

15 Curiosamente, incluso puede elegirse que la imagen de comunicación sea una imagen LDR de 100 nits (es decir, SDR), que queda lista de inmediato (sin necesidad de procesamiento adicional del color) para mostrarse en imágenes LDR antiguas (lo cual es una gran ventaja, porque

20 las pantallas antiguas no tienen conocimiento de HDR incorporado). ¿Cómo funciona eso? El televisor antiguo no reconoce los metadatos MDWPL (porque no existían en el estándar de video SDR, por lo que el televisor tampoco se dispone para buscarlos en algún lugar de la señal, p. ej.,

25 en un mensaje de información de mejora suplementaria, que es el mecanismo de MPEG para introducir todo tipo de

información técnica nueva previamente acordada). Tampoco
va a buscar la función. Simplemente examina la matriz de
color YCbCr, p. ej., de 1920 x 1080 píxeles, y muestra esos
colores como de costumbre, es decir, de acuerdo con la
5 interpretación de la Rec. 709 de SDR. Y el creador ha
elegido, en esta modalidad particular de códec, su función
FL_enc para que todos los colores, incluso la llama, se
mapeen a colores razonables en el rango SDR limitado. Nótese
que, en contraste con un simple cambio multiplicativo que
10 corresponde a la abertura o cierre del diafragma de una
cámara en una producción SDR (que típicamente conlleva un
recorte a al menos uno de los tonos blanco y/o negro), ahora
puede elegirse una forma de función óptima muy compleja,
siempre que se pueda revertir (p. ej., hemos enseñado
15 sistemas con una pregradación gruesa y después una
gradación fina). P. ej., puede ajustarse la luminancia
(respectivamente, brillo relativo) del auto a un nivel
apenas visible en SDR, p. ej., un negro profundo de 1 %,
mientras se ajusta la llama, p. ej., a 90 % (siempre que
20 todo siga siendo reversible). A primera vista, esto puede
parecer extremadamente abrumador, por no decir imposible,
pero numerosas pruebas de campo con todo tipo de material
de video y escenarios de uso han demostrado que, en la
práctica, es posible siempre que se haga correctamente
25 (siguiendo, p. ej., los principios de W02017157977).

¿Cómo sabemos ahora que se trata realmente de una señal de video HDR, incluso si contiene una imagen de color de píxeles utilizable por LDR, o de hecho que cualquier receptor compatible con HDR puede reconstruirla a HDR?

5 Porque también existen las funciones FL_dec en los metadatos, típicamente una por imagen. Y, por lo tanto, esa señal codifica lo que también desde el punto de vista colorimétrico, es decir, de acuerdo con nuestra discusión y definición anteriores, es una imagen HDR (5000 nits).

10 Aunque ya es más complejo que el sistema básico que solo transmite una imagen PQ-HDR, este código proxy DR sigue sin ser el mejor sistema preparado para el futuro, ya que sigue dejando que el lado receptor adivine cómo reducir los colores si tiene, p. ej., un televisor de 1500

15 nits, o incluso de 550 nits.

Por lo tanto, añadimos una visión técnica adicional, y desarrollamos la llamada tecnología de ajuste de pantalla (también conocida como adaptación de pantalla): la imagen puede sintonizarse para cualquier posible

20 televisor conectado, es decir, cualquier ML_D, porque puede duplicarse la función de la función de codificación FL_Enc como alguna función de guía para el mapeo ascendente de 100 nits IM_Comm no a una imagen reconstruida de 5000 nits, sino a, p. ej., una imagen de 1500 nits. La función cóncava,

25 que es sustancialmente la inversa de F_comp (nótese que, para el ajuste de la pantalla, no se requiere una inversión

exacta como sí se requiere para la reconstrucción), ahora
tendrá que escalarse para que sea algo menos pronunciada
(es decir, a partir de la función de decodificación de
referencia FL_dec se calculará una función de mapeo de
5 luminancia FL_DA adaptada a la pantalla), ya que ampliamos
a solo 1500 nits en lugar de 5000 nits. Es decir, puede
calcularse una imagen de rango dinámico terciario (DR_T),
p. ej., optimizada para una pantalla en particular, de
manera que la luminancia máxima de ese rango dinámico
10 terciario sea típicamente la misma que la luminancia máxima
que puede mostrar una pantalla en particular.

Las técnicas para esto se describen en WO2017108906
(podemos transformar una función de cualquier forma en una
función de forma similar que se encuentre más cercana a la
15 diagonal de 45 grados, en una cantidad que depende de la
relación entre las luminancias máximas de la imagen de
entrada y la imagen de salida deseada, vs. la relación de
las luminancias máximas de la imagen de entrada y una imagen
de referencia que aquí sería la imagen reconstruida, p. ej.,
20 usando esa relación para obtener puntos más cercanos para
todos los puntos de la diagonal, proyectando ortogonalmente
un segmento de línea desde el punto diagonal respectivo hasta
que se encuentre con un punto de la función de entrada; dichos
puntos más cercanos definen la función de salida ajustada,
25 que permite calcular las luminancias de la imagen Im_disp
que se va a mostrar a partir de las luminancias Im_comm.

No solo obtuvimos más tipos de pantallas, incluso para contenido básico de películas o televisión (televisores LCD, teléfonos móviles, proyectores de cine en casa, proyectores digitales profesionales para salas de cine), y más fuentes de video y medios de comunicación diferentes (satélite, transmisión por Internet, p. ej., OTT, transmisión por 5G), sino que también obtuvimos más formas de producción de video.

La Figura 2 muestra, generalmente, sin desear ser limitante, algunas creaciones típicas de video donde las presentes enseñanzas pueden desplegarse de manera útil.

En un entorno de estudio, p. ej., para las noticias o una comedia, todavía puede haber un entorno de filmación estrictamente controlado (aunque HDR permite relajar esto y filmar en entornos reales). Habrá iluminación controlada (202), p. ej., una batería de luces base en el techo, y diversos focos de luz. Habrá un número de cámaras de televisión voluminosas relativamente fijas (201). Las variaciones en esta transmisión frecuentemente en tiempo real serán, p. ej., un programa deportivo como fútbol, que tendrá diversos tipos de cámaras como cámaras cercanas al arco para una vista local, cámaras de visión general, drones, etc.

Habrá algún entorno de producción 203, en el que las diversas alimentaciones de las cámaras pueden seleccionarse para convertirse en la alimentación final, y pueden tomarse diversas decisiones de gradación (típicamente

simples, pero potencialmente más complejas). En el pasado esto sucedía frecuentemente en, p. ej., un camión de producción, que tenía muchas pantallas y diversos operadores, pero con flujos de trabajo basados en Internet, 5 donde las alimentaciones sin procesar pueden viajar a través de alguna red, la composición final puede ocurrir en las instalaciones de la emisora. Por último, al simplificar la producción para esta elucidación, parte de la codificación y formateo para la distribución de difusión a clientes 10 finales (o intermedios, tales como las estaciones de cable locales) sucederá en el formateador 204. Este típicamente hará la conversión a, p. ej., PQ YCbCr a partir de las luminancias según se gradaron como se explicó con la Figura 1, para, p. ej., un formato de rango dinámico intermedio, 15 calculará y formateará todos los metadatos necesarios, convertirá a algún formato de difusión como DVB o ATSC, empaquetará la información en trozos para distribución, etc. (etc. indica que puede haber tablas añadidas para señalar contenido disponible, subtítulo, cifrado, pero al menos 20 parte de eso lo hará ser de menor interés para comprender los pormenores de las innovaciones técnicas actuales).

En el ejemplo, el video (una transmisión de televisión en el ejemplo) se comunica mediante un satélite de televisión 250 a una antena parabólica 260 y a un 25 decodificador con capacidad para señal satelital 261. Finalmente se mostrará en una pantalla de usuario final 263.

Esta pantalla puede estar mostrando este primer video, pero también puede mostrar otra alimentación de video, incluso potencialmente al mismo tiempo, p. ej., en ventanas de imagen en imagen (o algunos datos del primer programa de video HDR pueden venir por medio de algún mecanismo de distribución y otros datos por medio de otro).

Una segunda producción es típicamente una producción fuera de línea. Podemos pensar en una película de Hollywood, pero también puede ser un espectáculo de alguien que corre a través de una jungla. Tal producción puede filmarse con otras cámaras óptimas, p. ej., la steadicam 211 y el dron 210. Nuevamente suponemos que las alimentaciones de la cámara (que pueden estar sin procesar, o ya convertidas a algún formato de producción HDR como HLG) se almacenan en algún lugar en la red 212, para procesarse más tarde. En tal producción podemos pedir, en los últimos meses de producción, a algún técnico humano de gradación de color que use el equipo de gradación 213 para determinar las luminancias óptimas (o brillos relativos en caso de producción y codificación de HLG) de la gradación maestra. Después, el video puede subirse a algún servicio de video basado en Internet 251. Para la distribución de video profesional, esto puede ser, p. ej., Netflix.

Un tercer ejemplo es la producción de videos de consumo. Aquí el(la) usuario(a) tendrá, p. ej., al hacer un vlog, un encendedor de anillo 221, y hará la captura

por medio de un teléfono móvil 220, pero también puede estar haciendo la captura en alguna ubicación exterior sin iluminación suplementaria. Él(Ella), típicamente, subirá el video a Internet, quizás a YouTube, TikTok, etc.

5 En el caso de recepción mediante el Internet, la pantalla 263 estará conectada mediante un módem, enrutador 262 o lo similar (configuraciones más complejas como redes wifi domésticas no se muestran en esta explicación).

Otro usuario puede estar viendo el contenido de
10 video en una pantalla portátil (271), tal como un computador (o, similarmente, otros usuarios pueden usar una PC de escritorio no portátil), un teléfono móvil, etc. Pueden acceder al contenido a través de una conexión inalámbrica (270), tal como Wi-Fi, 5G, etc.

15 Así pues, puede observarse que hoy en día pueden generarse y comunicarse diversos tipos de video, con diferentes codificaciones técnicas, de diversas maneras, y nuestros sistemas de codificación y procesamiento se han diseñado para manejar prácticamente todas esas variantes.

20 La **Figura 3** muestra un ejemplo de un circuito de conversión del rango dinámico absoluto (definido por nivel nit) 300 para un decodificador de imagen o video (HDR) (el codificador funciona de manera similar, pero típicamente con funciones invertidas, es decir, la función que se va
25 a aplicar es la función del otro lado reflejada en la diagonal). Se basa en codificar una imagen primaria (p.

ej., una gradación HDR maestra) con un rango dinámico de luminancia primaria (DR_Prim) como otra imagen (denominada proxy) con un rango secundario diferente de luminancias de píxeles (DR_Sec). Si el codificador y todos sus
5 decodificadores compatibles han acordado previamente o saben que la imagen proxy tiene una luminancia máxima de 100 nits, no es necesario comunicarlo como un metadato SDR_WPL. Si la imagen proxy es, p. ej., una imagen con un máximo de 200 nits, esto se indicará rellorando su
10 luminancia de punto blanco proxy P_WPL con el valor 200, o de manera similar para 80 nits, etc. El valor máximo de la imagen primaria (HDR_WPL= 1000), que será reconstruido por el circuito de conversión del rango dinámico, normalmente se comunicará como metadatos de la imagen de
15 entrada recibida o de la señal de video, es decir, junto con los tripletes de color de los píxeles de entrada (Y_in, Cb_in, Cr_in). Las distintas luma de los píxeles llegarán típicamente como un plano de imagen de luma; es decir, los píxeles secuenciales tendrán la primera luma Y11, la
20 segunda Y21, etc. (típicamente estos se escanearán, y el circuito de conversión del rango dinámico convertirá píxel por píxel en tripletes de color de píxeles de salida (Y_out, Cb_out, Cr_out). En esta explicación, nos centraremos principalmente en la dimensión de brillo de
25 los colores de los píxeles. Los distintos circuitos de conversión de rango dinámico pueden funcionar internamente

de manera diferente para lograr básicamente lo mismo: una luminancia de salida L_{out} correctamente reconstruida para todos los píxeles de la imagen (los detalles reales no importan para esta innovación, y las modalidades se centrarán en enseñar solo los aspectos necesarios).

El mapeo de luminancias del rango dinámico secundario al rango dinámico primario puede ocurrir en las luminancias mismas, pero también en cualquier representación de luma (es decir, de acuerdo con cualquier EOTF u OETF), siempre que se haga correctamente, p. ej., no por separado en componentes $R'G'B'$ no lineales. La representación interna de luma ni siquiera tiene por qué ser la de la entrada (es decir, de Y_{in}), ni tampoco la de cualquier salida que puedan proporcionar los sistemas de circuitos de conversión del rango dinámico o el decodificador que los integra (p. ej., una luma de formato Y_{sigfm} para un formato o sistema de comunicación particular, que se “comunica” incluyendo el almacenamiento en una memoria, p. ej., dentro de una PC, un disco duro, un medio de almacenamiento óptico, etc.).

Hemos mostrado opcionalmente (con puntos) un circuito de conversión de luma 301, que transforma las lumas de entrada Y_{in} en lumas uniformizadas desde el punto de vista perceptivo Y_{pc} .

El solicitante ha normalizado en ETSI 103433 una ecuación útil para convertir luminancias de cualquier rango a tal representación perceptual de la luma:

$$Y_{pc} = \log_{10}\{1 + [\text{RHO}(\text{WPL}_{inrep}) - 1] * \text{power}(\text{Ln}_{in}; 1/(2,4))\} / \log_{10}\{\text{RHO}(\text{WPL}_{inrep})\}$$

5 En la que la función RHO se define como

$$\text{RHO}(\text{WPL}_{inrep}) = 1 + 32 * \text{power}\{(\text{WPL}_{inrep}/10000); 1/(2,4)\}$$

El valor WPL_{inrep} es la luminancia máxima del
 10 rango que debe convertirse a lumas uniformizadas psicovisualmente; así, para una imagen SDR de 100 nits, este valor sería 100, y para la imagen de salida que se va a reconstruir (o la imagen codificada originalmente en el lado de creación), el valor sería 1000.

15 Ln_{in} son las luminancias a lo largo de cualquier rango que necesite convertirse, después de la normalización dividiéndolas por su luminancia máxima respectiva, es decir, dentro del rango [0,1].

Una vez que tenemos un rango de entrada y salida
 20 normalizado a 1,0, podemos aplicar una función de mapeo de luminancia en el dominio de la luma, como se muestra dentro del circuito de mapeo de luma 302, que realiza el mapeo de luma real para cada píxel entrante.

De hecho, esta función de mapeo había sido elegida
 25 específicamente por el codificador de la imagen (al menos para obtener una buena calidad de reconstrucción, y tal vez

también una cantidad reducida de bits necesarios al comprimir MPEG, pero a veces también para cumplir con otros criterios como, p. ej., que la imagen proxy SDR tenga la distribución de luminancia correcta para la escena en particular (una cueva oscura o una explosión diurna) en una pantalla SDR tradicional, etc.). Por lo tanto, esta función F_{dec} (o su inversa) se extraerá de los metadatos de la señal o representación de la imagen de entrada y se suministrará al circuito de conversión del rango dinámico para realizar el mapeo de luma real por píxel. En este ejemplo, la función F_{dec} especifica directamente el mapeo necesario en el dominio de luma perceptual, pero por supuesto son posibles otras variantes, ya que también pueden aplicarse diversas conversiones a las funciones. Además, aunque por simplicidad en la explicación y para garantizar la comprensión de la enseñanza, aquí enseñamos una conversión del rango dinámico del decodificador puro, otras conversiones del rango dinámico pueden usar otras funciones, p. ej., una función derivada de F_{dec} , etc. Los detalles de todo ello no son necesarios para comprender la presente contribución innovadora a la tecnología.

En general, no solo se modificarán las luminancias, sino que también se producirá un cambio correspondiente en los cromas Cb y Cr. Esto también puede lograrse de varias maneras, desde la decodificación estrictamente inversa hasta la implementación de

características adicionales como un aumento de la saturación, ya que Cb y Cr codifican la saturación de los píxeles. Para ello, típicamente se comunica otra función en los metadatos (función de especificación de recoloración FCOL), que determina el comportamiento de recoloración cromática, es decir, el mapeo de Cb y Cr (nótese que Cb y Cr típicamente se cambiarán en la misma cantidad multiplicativa, ya que la relación Cr/Cb determina el tono, y generalmente no se desean cambios de tono al decodificar, es decir, la imagen de rango dinámico inferior y superior generalmente tendrá píxeles de objeto con diferente brillo, y a menudo al menos algunos de los píxeles tendrán diferente saturación, pero idealmente el tono de los píxeles en ambas versiones de la imagen será el mismo). Esta función de color típicamente especificará un multiplicador cuyo valor depende de un código de brillo Y (p. ej., Y_{pc} u otros códigos en otras variantes). Un circuito de establecimiento del multiplicador 305 proporcionará el multiplicador m adecuado para la situación de brillo del píxel que se está procesando. Un multiplicador 306 multiplicará tanto Cb_{in} como Cr_{in} por este mismo multiplicador, para obtener los cromas de salida correspondientes $Cb_{out} = m * Cb_{in}$ y $Cr_{out} = m * Cr_{in}$. Por lo tanto, el multiplicador realiza el procesamiento cromático correcto, por lo que todo el procesamiento del color de cualquier conversión de rango

dinámico puede configurarse correctamente en el circuito de conversión del rango dinámico.

Además, típicamente puede haber (al menos en un decodificador) un circuito de formato 310, de manera que el triplete de color de salida (Y_{out} , Cb_{out} , Cr_{out}) pueda 5 convertirse a cualquier formato de salida necesario (p. ej., un formato RGB o un formato de comunicación YCbCr, Y_{sigfm} , Cb_{sigfm} , Cr_{sigfm}). P. ej., si el circuito emite a una versión de un canal de comunicación 379 que es un 10 cable HDMI, tales cables típicamente usan la codificación de color de píxeles YCbCr basada en PQ, por lo tanto, las lumas se convertirán nuevamente del dominio perceptual al dominio PQ por el circuito de formato.

Es importante que el lector comprenda bien qué 15 es una (de)codificación, y en qué se diferencia una imagen HDR absoluta, o sus colores de píxeles, de una imagen SDR antigua. Puede haber una conexión con un circuito de ajuste de pantalla 380, que calcula los colores y la luminancia finales de los píxeles que se mostrarán en la pantalla de 20 algún dispositivo, p. ej., un televisor de 450 nits que algún consumidor tenga en casa.

Sin embargo, en HDR absoluto, pueden establecerse las luminancias de los píxeles ya en la etapa de decodificación, al menos para la imagen de salida (en 25 este caso, la imagen de 1000 nits).

Lo hemos mostrado en la Figura 3B, para una imagen HDR típica que sea una imagen de interior/exterior. Nótese que, si bien en el mundo real la luminancia exterior suele ser 100 veces más brillante que la interior, en una
5 imagen HDR masterizada real puede ser mejor aumentarla, p. ej., 10 veces, ya que el espectador la verá toda en una pantalla, con un ángulo de visión fijo, incluso típicamente en una habitación con iluminación tenue por la noche, y no en el mundo real.

10 Sin embargo, encontramos que cuando miramos las luminancias correspondientes a las luma, p. ej., las luminancias HDR L_{out} , típicamente vemos un gran histograma (de recuentos $N(L_{out})$ de cada luminancia que aparece en una imagen de salida de esta escena hogareña). Esto se extiende
15 considerablemente por encima de algunos lóbulos de rango dinámico más bajos de luminancias, y por encima del nivel de rango dinámico bajo de 100 nits, porque las imágenes de exteriores soleados tienen su propio lóbulo de histograma. Obsérvese que la representación de luminancia se dibuja de
20 manera no lineal, p. ej., logarítmicamente. También podemos rastrear lo que haría el codificador en el lado de la codificación, al crear la imagen proxy de 100 nits (y su histograma de recuentos de luminancia proxy $N(L_{in})$). Se usa una función convexa, como se muestra en la Figura 1, o dentro
25 del mapeador de luma 302, que comprime las luminancias más brillantes, debido a las limitaciones del rango dinámico de

luminancia más pequeño. Aún existe cierta diferencia entre el brillo de interiores y exteriores, y un rango considerable para el lóbulo superior de los objetos exteriores, de manera que es posible crear los diferentes colores necesarios para

5 pintar los diversos objetos, tales como los diferentes tonos de verde en el árbol. Sin embargo, esto también implica algunos sacrificios. En primer lugar, los objetos de interior se mostrarán (suponiendo por el momento que una pantalla de 100 o 200 nits muestre fielmente esas luminancias tal como

10 están codificadas, y no, p. ej., realice algún procesamiento de embellecimiento arbitrario que las aclare) más oscuros, más oscuros de lo idealmente deseado, es decir, hasta el umbral de luminancia de interior T_{in} de la imagen HDR. En segundo lugar, la extensión del lóbulo superior se ve

15 reducida, lo que puede disminuir el contraste de los objetos exteriores. En tercer lugar, dado que los colores brillantes en la gama cromática en forma de tienda de campaña que se muestra en la Figura 1 no pueden tener una gran saturación, los colores exteriores pueden presentar un tono algo pastel,

20 es decir, con una saturación reducida. Pero, por supuesto, si el evaluador encargado de la creación tiene control sobre todas las funciones (F_{enc} , $FCOL$), puede equilibrar esas características, de manera que algunas presenten una menor desviación en detrimento de otras. P. ej., si el cielo

25 exterior es de un azul uniforme, el evaluador puede optar por hacerlo más brillante, aunque menos azul. Si se trata de

una hermosa puesta de sol, es posible que desee conservar todos sus colores y, en cambio, atenuar la iluminación, en particular si no hay rincones oscuros importantes en la parte interior de la imagen, ya que su contenido sería difícil de ver, especialmente al ver la televisión con todas las luces encendidas (nótese que también existen técnicas para manejar las diferencias de iluminación y la visibilidad de los negros, pero esa es demasiada información para la explicación de esta solicitud de patente).

El gráfico central muestra cómo se verían las lumas para las luminancias proxy, lo que normalmente puede dar un histograma más uniforme, con, p. ej., aproximadamente el mismo rango para las luminancias de los objetos de la imagen en interiores y exteriores. Sin embargo, las lumas solo son relevantes en la medida en que codifican las luminancias, o en caso de que se realicen algunos cálculos en el dominio de la luminancia (lo que tiene ventajas para el tamaño de la longitud de palabra en los circuitos de procesamiento). Nótese que, si bien el formalismo absoluto puede asignar luminancias en el lado de entrada entre cero y 100 nits, también pueden tratarse las luminancias SDR como brillos relativos (que es lo que haría una pantalla antigua, al descartar todo el conocimiento HDR y los metadatos comunicados, y considerar únicamente los códigos de luma y croma de 0-255).

La **Figura 4** ilustra cómo el mundo de la informática consideraba el HDR, en particular para el cálculo de escenas HDR, p. ej., mediante el trazado de rayos (es decir, el equivalente a la captura real con una cámara). Esta tecnología, que puede usarse, p. ej., en videojuegos, en los que es necesario recalcular la vista de un entorno diferente cada vez que un jugador se desplaza, lo que puede dar lugar a que aparezca un reflejo especular que no se veía cuando el jugador se encontraba un metro de juego a la izquierda, no fue diseñada principalmente para la comunicación, p. ej., de una emisora a un receptor (cualquier receptor) con cualquiera de los diversos tipos de pantallas con diferentes características de brillo máximo. Estos cálculos y representaciones se crearon para una aplicación típica (estrictamente controlada) dentro de una única PC, y ciertamente no se desarrollaron teniendo en cuenta muchas aplicaciones futuras muy diferentes. Cuando se calcula una imagen HDR, p. ej., definiendo una fuente de luz muy brillante y modelando físicamente (internamente) cómo ilumina un píxel de un espejo, puede definirse cualquier luminancia HDR. Pero uno de los problemas es que ya puede calcularse, pero no necesariamente mostrarla. P. ej., típicamente se usaría una representación en punto flotante (p. ej., medio flotante de 16 bits para representar las luminancias en un formato logarítmico, es decir, capaz de representar luminancias casi infinitas, básicamente ridículamente grandes desde el punto

de vista de su uso real para visualización). Se considerarían todas las luminancias inferiores a 1 (que también puede expresarse como el 100 % relativo) como las luminancias SDR normales. Puede que no sea la mejor manera de tratar, p. ej., la visualización de una imagen HDR flotante, pero algunos sistemas, componentes o software recortarían todo lo que esté por encima de 1 a 1,0, y simplemente usarían las luminancias más bajas. Los objetos exteriores, como la casa, cuando se generan por computador, pueden tener luminancias con un factor de multiplicación superior a 1,0, p. ej., 5 o 20 veces mayor (si se asociara 1,0 con 100 nits, lo cual no necesariamente se hizo en la vista del computador, se podría decir que serían, por ejemplo, 500 nits). Se podría hacer que el sol, p. ej., brillara hasta 1 millón de nits, lo que necesitaría un mapeo descendente severo para que fuera visible como una esfera de color amarillo claro, o, típicamente, se recortaría a blanco (lo cual no sería un problema, ya que en el mundo real el sol también es tan brillante que se recortará a blanco en las capturas de las cámaras). Sin embargo, puede haber problemas con otros objetos en el mapeo descendente.

Una diferencia que ya se observa, p. ej., con la representación PQ, es que esta representación es cerrada (se asume que cualquier luminancia que se necesite en la práctica puede razonablemente caer por debajo de su máximo de 10.000 nits), y por lo tanto pueden realizarse, p. ej.,

mapeos de luminancia basados en este punto final, mientras que el formato log informático es, pragmáticamente, abierto, en el sentido de que puede ir casi infinitamente por encima de 1. Se podría decir que incluso para un formato logarítmico de 16 bits existe un punto final, pero es mucho menos interesante mapearlo que mapear los valores alrededor de 1,0 (comprimir, p. ej., linealmente desde 1 millón de nits haría que todos los píxeles interiores fueran completamente negros). Por lo tanto, las representaciones de rango abierto necesitan un mapeo de luminancia algo diferente (denominado mapeo de tonos en esta subárea), pero en un nivel más general puede llegarse a un enfoque común, al menos en el sentido de que ambos subcampos de la tecnología han podido crear imágenes LDR (y similarmente, en el ámbito moderno, también pueden crearse imágenes HDR (comunes) para ambos).

Los videojuegos, que hasta hace poco se jugaban mayoritariamente en pantallas SDR, a pesar de producir imágenes bellas y realistas, necesitaban mapear descendente esas pantallas para obtener imágenes SDR. Se podría argumentar que eso no sería muy diferente del mapeo descendente de imágenes HDR naturales capturadas por cámaras, algo que no era habitual hasta hace poco. Pero los mapeadores de tonos a veces eran difíciles de comprender y un tanto improvisados. Sin embargo, los videojuegos en HDR se hicieron realidad, aunque con algunos problemas iniciales, y a la gente le encantó el aspecto. Pero el hecho

de que se hayan resuelto algunos problemas, como en el caso de la televisión HDR, otra aplicación estrictamente controlada, no significa que todos los problemas de HDR estén resueltos en el mundo de la informática.

5

Televisión versus entorno informático

Aunque todavía existen muchos tipos (y visiones técnicas, p. ej., la codificación absoluta basada en PQ vs. la codificación relativa basada en HLG), como se muestra
10 anteriormente, al menos para la comunicación de video simple (p. ej., para los servicios de televisión por difusión), la situación es relativamente simple y similar a la situación de la “línea de control de pantalla directa” de la era analógica PAL (que usaba el paradigma de visualización: el
15 código más brillante de la imagen, el blanco, se muestra como lo más brillante en la pantalla, y el espectador lo percibe como el blanco más blanco). Al menos se recibe un video que se entiende bien (porque está estandarizado), y simplemente necesita decodificarse, y mostrarse como un único elemento
20 en toda la pantalla (no necesariamente solo como una ventana en la pantalla del televisor, p. ej., una página web).

El entorno informático en contraste es un entorno muy heterogéneo. En la década de 1980, casi todo estaba muy enfocado en computadores específicos. Existían
25 versiones propietarias (simplificadas) de software, y dado que a algunos computadores les faltaban muchas

funcionalidades, esto no hacía que el computador de propósito general fuera tan general en realidad. Un comprador privado tenía que conformarse con el computador que compraba, y algunos vendedores de computadores en los años 90 vendían el computador junto con un paquete de aplicaciones de software útiles, tal como una enciclopedia en CD-ROM, etc. Las empresas podían optar por comprar computadores Apple para todos sus empleados o computadores con Windows para PC, y la cooperación o la subcontratación laboral en aquella época no eran tan comunes como hoy en día. P. ej., el software de procesamiento de textos de Tandy simulaba de manera rudimentaria los menús de selección dibujando unas pocas líneas alrededor de una lista vertical de palabras clave. Esas aplicaciones tenían que tomar el control de toda la pantalla, mientras que los sistemas operativos modernos pueden manejar diversas aplicaciones en varias ventanas (la complicada situación actual del HDR parece un retroceso a los años 80).

Aunque los monitores de tubo de rayos catódicos que cada cliente tenía en su configuración informática específica eran, en principio, de la misma construcción, los formatos de conectores de cable específicos hacían que todo fuera único. Incluso el orden de los bytes de los datos variaba entre el mundo de Apple y el de las PC, lo que dificultaba aún más el intercambio. Por un lado, eso facilitaba las cosas, ya que un programador de Macintosh

podía estudiar uno, y más tarde tres, MacBooks y saber todo lo que necesitaba, pero por otro lado, un desarrollador de, p. ej., un juego, necesitaba crear muchas versiones diferentes para las distintas plataformas si quería llegar a un público más amplio. Esto provocó, entre otras cosas, que incluso a principios del siglo XXI los usuarios de Macintosh no pudieran realizar operaciones bancarias por Internet con el software específico proporcionado por el banco.

Esto mejoró cuando tal proceso de banca por Internet pasó a basarse en servidores. En aquel entonces, cualquier computador solo necesitaba implementar adecuadamente los elementos de protocolo estandarizados, como p. ej. un navegador para comunicarse mediante HTTPS, leer y mostrar código HTML, etc. Para la comunicación en “texto plano”, eso siempre parecería funcionar, al igual que se podrían usar formatos altamente estandarizados como pdf o jpeg para la comunicación y cooperación entre sistemas. Sin embargo, en el fondo, todavía existían muchas diferencias. P. ej., si bien los distintos sistemas tenían una interfaz de usuario visual con ventanas, y todos ellos contaban de manera similar con un área de imagen en la que una aplicación podía dibujar algo, una barra de desplazamiento para desplazarse si el tamaño del contenido era mayor que el área de la ventana, y una barra para seleccionar y colocar la ventana, no solo el aspecto de, p. ej., una ventana X-windows para Unix era diferente al de

Mac, sino, lo que es más importante, la manera de definirla. Además, aunque los sistemas Mac, PC y Unix creaban colores RGB para un mismo monitor CRT (o un monitor LCD que, por su hardware interno, emulaba el comportamiento de visualización de un monitor CRT), esto conllevaba a que los colores gráficos predefinidos (también conocidos como paleta) de X-
5 Windows no fueran los mismos que los de un Mac o una PC.

En la práctica, con la rápida evolución de los computadores, que se convirtieron más en un conjunto de
10 piezas, casi todo cambió (excepto la convergencia en la estandarización de los cables a USB, HDMI y lo similar). El mismo computador (aparato) podría ejecutar Windows (y de hecho, cualquier versión aún más antigua de Windows), o Linux. Pero incluso si el propietario del computador hubiera
15 instalado una versión común de Windows, el hardware del computador podría cambiar: p. ej., podría tener una unidad de procesamiento de gráficos (GPU) de AMD, Intel o Nvidia (y dado que los teléfonos inteligentes se han convertido en computadores populares para la situación actual, podemos
20 añadir los populares procesadores Qualcomm). Y algunos de ellos podrían tener, p. ej., 2 GB de memoria RAM de video (VRAM, por sus siglas en inglés) para almacenar, entre otras cosas, texturas en la GPU, mientras que otros podrían tener 8 GB. Para las aplicaciones web, puede optarse por
25 conectarse a Internet mediante navegadores, como p. ej., Edge, Chrome, Firefox (o el navegador específico de Samsung

en los teléfonos Samsung; y diversas aplicaciones en televisores), etc. Las GPU tenían su propia funcionalidad, los sistemas operativos tenían su comportamiento predefinido, y el creador de una aplicación simplemente
5 quería insertar una frase de texto escrito o una imagen.

Los problemas que se describen en la presente y las soluciones que ofrecen las modalidades serán similares tanto para las aplicaciones nativas (escritas específicamente para una plataforma informática específica,
10 y que se ejecutan en ella) como para las aplicaciones web actualmente populares, que se sirven desde un servidor remoto, es decir, que tienen varios servicios de ese servidor remoto, pero que pueden depender para algunos cálculos (p. ej., ejecutar código JavaScript descargado)
15 en el computador del cliente local, y pueden necesitar hacerlo para determinadas etapas del programa. Para el cliente, no parece haber mucha diferencia entre que su hoja de cálculo esté instalada localmente o ejecutándose en la nube (excepto quizás por la cuota de suscripción). Es
20 posible que ni siquiera se dé cuenta de que, si está escribiendo un correo en Gmail ejecutándose en un navegador, en realidad, bajo la superficie, está haciendo uso de tecnologías de Internet, ya que está escribiendo líneas de HTML. La diferencia entre un explorador de
25 archivos local y un explorador de archivos de Internet es cada vez menor. Por otro lado, dado que cualquier servicio

se ejecuta a través de Internet entonces, esos sistemas pueden querer hacer uso de características cada vez más avanzadas, que pueden necesitar recurrir a detalles del dispositivo informático del cliente, bajo el capó. P. ej., un juego de Internet controlado por un servidor puede usar una granja de procesadores para calcular el comportamiento de los actores virtuales, pero quiere beneficiarse de la aceleración de hardware de los cálculos en la GPU del cliente, p. ej., para hacer el sombreado final (con sombreado en el sentido informático nos referimos a los cálculos necesarios para llegar a los colores correctos, que incluye el brillo de un objeto, p. ej., usando una simulación de iluminación de algún objeto con alguna textura física como una tapicería, o interpolación de colores de los vértices de un triángulo, cuyos colores de píxeles típicamente terminan como componentes de color rojo, verde y azul en un búfer de color; también puede usarse el término renderizado, que se refiere al proceso completo de nivel superior para obtener una imagen de un objeto desde un punto de vista, para cualquier generación de una matriz NxM de píxeles en un llamado lienzo, que es una memoria en la que se colocan los píxeles finalmente renderizados, teniendo en cuenta también aspectos como la visibilidad y la oclusión. Si bien no queremos implicar ninguna limitación, explicaremos algunos detalles técnicos

de nuestras innovaciones con modalidades de aplicaciones web más complejas.

Existen problemas particulares cuando es necesario mostrar de manera coordinada varios elementos visuales con diferentes rangos dinámicos de brillo, tal como p. ej. en un sistema de ventanas de un computador (y posiblemente resulte más complicado en varias pantallas con capacidades de brillo posiblemente diferentes). Esto es un problema técnico de por sí complejo, y el hecho de que, en la práctica, intervengan muchos componentes diferentes y, por lo tanto, productores/empresas diferentes (diversas aplicaciones de software o middleware, diferentes sistemas operativos controlados por empresas como Microsoft o Google, diferentes proveedores de unidades de procesamiento gráfico, etc.) no necesariamente facilita las cosas. En la era de Internet, cualquiera puede crear contenido visual, desde una persona particular en Filipinas hasta un profesional que gestiona una página web creada y alojada en los Estados Unidos y destinada a visitarse al menos en ese país, y puede ser necesario mezclar y presentar de manera coordinada varios de estos contenidos al mismo tiempo. En este texto, con **elemento visual** no nos referimos necesariamente a un elemento que puede mostrarse en un área que ha sido preparada previamente, p. ej., almacenada en una memoria, sino también a elementos visuales que pueden generarse sobre la marcha, p. ej., un texto con colores HDR generados a medida que se escribe en

alguna ventana, y que terminan en algún lienzo que recopila todos los elementos visuales de las distintas ventanas, para finalmente ser mostrados después de recorrer toda la tubería de procesamiento que se configura en el computador, con ese conjunto particular de aplicaciones que presentan sus elementos visuales. Varios elementos visuales pueden corresponder a diversas ubicaciones geométricas de una imagen y/o colocarse en diversas ubicaciones en alguna imagen o espacio reservado (memoria) de una imagen como un lienzo. El flujo de manejo (posiblemente procesamiento, pero no totalmente fuera del control del creador de elementos) de los elementos de imagen terminará con los píxeles finales en el búfer de salida de escaneo, los cuales pueden ser mostrados en una o más pantallas conectadas (posiblemente la pantalla también pueda seguir realizando procesamiento de color, que incluye la luminancia, pero otras variantes pueden simplemente producir colores de imagen para una visualización directa tal como están definidos esos colores).

Para ser claros, lo que entendemos por unidad de procesamiento de gráficos es un circuito (o tal vez en algunos sistemas circuitos, si hay dos o más GPU separadas a las que se les suministran los elementos visuales que se mostrarán en su totalidad) que contiene el **almacenamiento final** para tales píxeles que deben mostrarse en una pantalla, es decir, ser escaneados a al menos una pantalla (y no procesadores que no tienen este búfer de salida de escaneo y sistemas de

circuitos para comunicar las señales de video a la(s) pantalla(s); esta GPU a veces también se denomina unidad de procesamiento de pantalla (DPU, por sus siglas en inglés). Por lo tanto, no se trata de una memoria de almacenamiento temporal para realizar un cálculo intermedio.

Por lo tanto, es necesario un enfoque universal y bien coordinado para gestionar los elementos visuales HDR (y posiblemente algunos SDR) en computadores, que pueden tener varias aplicaciones ejecutándose en paralelo, ninguna de las cuales controla toda la pantalla visible, y que pueden funcionar (en el sentido de al menos enviar sus colores de píxeles preferidos al búfer de salida de escaneo de la GPU) a través de diversas capas de software, middleware, API, etc.

<https://mpv.io/manual/stable> de 27 noviembre 2023 es el manual de medios mpv. Especifica un número de opciones de renderizador de GPU de línea de comandos.

Rod Lawton
URL:<https://www.lifeafterphotoshop.com/local-adjustments-explained/> instruye sobre los ajustes locales basados en máscara (en Lightroom o Capture One) para fotos, por ej., para oscurecer el cielo.

Victoria Bampton: “Lightroom Performance - Previews & Caches”
<https://www.lightroomqueen.com/lightroom-performance-previews-caches/> resume las vistas previas JPEG que pueden almacenarse para sus fotos RAW, y versiones procesadas

parcialmente en caché (resolución más baja) de archivos accedidos recientemente para agilizar los tiempos de carga.

Dai Min y Dmytro Rusanovskyy: “An overview of end-to-end HDR,” Proceedings of the SPIE,

5 vol. 10752, 17 septiembre 2018 ofrece una síntesis de las técnicas de HDR desde la captura a EOTF, procesamiento, estándares y mapeo de tonos.

BREVE DESCRIPCIÓN DE LA INVENCION

10

Los problemas indicados se resuelven mediante un computador (700) que ejecuta al menos una aplicación de software primaria (701), en la que la aplicación de software primaria se dispone para presentar, con el fin de

15 mostrarlos, elementos visuales de diferentes rangos de brillo, en donde la aplicación de software primaria presenta un primer elemento visual (702) que comprende píxeles cuyos brillos (Br_H) se encuentran dentro de un primer rango de brillo (DR_H);

20 en donde la aplicación de software primaria comprende un mapeador de luma (704) dispuesto para mapear un primer triplete de código de color (RGB1) que codifica un primer brillo (Br_H) de un píxel (7201) a un segundo triplete de código de color correspondiente (RGB2) que codifica un

25 segundo brillo (Br_L) para dicho píxel, aplicando una o más funciones de mapeo primarias (F_Y) al primer triplete

de código de color que produce el segundo triplete de código de color, en donde el segundo brillo es un brillo que se encuentra en un segundo rango de brillo (DR_L) de un tamaño diferente al del primer rango de brillo;

5 los tripletes de color primero y segundo representan colores asociados en un rango dinámico de luminancia considerablemente diferente, p. ej., la luminancia máxima o el brillo que se va a mostrar del primer brillo/rango dinámico puede ser para varios elementos típicamente al

10 menos 5 veces mayor que el del segundo rango dinámico; en donde la aplicación de software primaria comprende un formato (707) dispuesto para componer segundos tripletes de código de color (RGB2) a partir del mapeo de luma realizado por el mapeador de luma en una primera estructura de datos

15 de imagen (IMDAT, por sus siglas en inglés) para comunicarse con un software secundario (720), que típicamente se formateará de una manera que facilite la comunicación entre diferentes software, típicamente mediante alguna interfaz de programación de aplicaciones, y comunicará al software

20 secundario (720) la primera estructura de datos de imagen (IMDAT) **junto con** una codificación de una o más funciones de mapeo primarias (F_Y) y una codificación de una región espacial (REG1) de aplicabilidad de una o más funciones de mapeo primarias (F_Y), en donde la una o más funciones de

25 mapeo primarias son reversibles de manera que el primer triplete de código de color (RGB1) resulta de aplicar la

inversa de una o más funciones de mapeo primarias en el segundo triplete de códigos de color (RGB2);

(la codificación de la función de mapeo HDR a LDR puede representarse de diversas maneras dependiendo de la función que se use, p. ej., si es una función polinómica, las constantes de multiplicación para las distintas potencias en la suma pueden comunicarse junto con los demás datos, en la llamada a la API que transfiere la imagen HDR en cualquier sentido, es decir, enviando activamente o llamando, etc.);

10 en donde la aplicación de software primaria comprende un controlador primario (710) dispuesto para generar una señal de control **de paso directo** (C_PT) que instruye a un controlador secundario (722) en el software secundario (720) que una o más funciones de mapeo primarias (F_Y), o

15 una o más funciones de mapeo derivadas (F_Y*) que se basan en la una o más funciones de mapeo primarias (F_Y), deben pasarse a un **software terciario (740)**;

en donde el software terciario controla el suministro de colores de píxeles finales para un búfer de salida de

20 escaneo (511) de una unidad de procesamiento de gráficos GPU (510), en donde al menos algunos de los colores finales son copias de los segundos tripletes de color, y en donde el software terciario se dispone para pasar la una o más funciones de mapeo primarias (F_Y) respectivamente o una

25 o más funciones de mapeo derivadas (F_Y*) a la GPU.

La aplicación de software primario típicamente es una aplicación que presta servicio al usuario. Por lo tanto, típicamente será un software que se esté ejecutando. Sus partes, como p. ej. un mapeo de luma, serán típicamente algoritmos o conjuntos de rutinas de software, es decir, una o más etapas de un método (incluso si el computador que ejecuta el software es en realidad un aparato con una constitución y un comportamiento de funcionamiento específicos, ya que en nuestra era digital parte del comportamiento de los computadores es controlable en realidad por variantes del software en ejecución). P. ej., el usuario puede estar interactuando con una página web dinámica mediante un navegador, y hoy en día las páginas web pueden tener diversas regiones que intentan interactuar con el usuario, o incluso captar su atención. P. ej., puede haber ventanas o áreas comerciales que se carguen sobre la marcha, las cuales no deberían ser tan intrusivas como para que el usuario no pueda encontrar lo que realmente estaba buscando, p. ej., la casilla de consulta donde escribiría el nombre de un libro para pedir. Sin embargo, la armonización del brillo de los distintos elementos visuales dentro o fuera de la ventana del navegador es solo una técnica secundaria, ya que las presentes modalidades se centran principalmente en poder obtener cualquier brillo de píxel deseado correctamente en el búfer de salida de escaneo. Si no se aplicara el enfoque

innovador actual, los colores de los píxeles que viajan entre un software y otro pueden, típicamente, sufrir de manera impredecible y sin control indirecto varios mapeos de rango dinámico (algunos de los cuales podrían ser muy rudimentarios y, p. ej., recortar partes importantes del elemento), y también podrían ser ad hoc y, p. ej., realizar alguna mejora genérica a un rango dinámico más alto que puede cambiar considerablemente los colores en comparación con lo que el creador del elemento desearía (y después tal vez otra degradación en un siguiente software que maneje el elemento o los píxeles que se generaron a partir del elemento, etc.). Con el método actual puede controlarse de forma más estable tanto cómo debería mapear idealmente los colores del recurso cualquier procesamiento posterior, como que pueda solicitarse que un procesamiento final, p. ej., para adaptarlo a una pantalla, se realice en una unidad de procesamiento de gráficos (GPU). Puede observarse que este método de comunicación/coordinación de color HDR, al añadir las funciones referidas a una representación de referencia común (p. ej., sRGB, en la que el blanco puede representar algún nivel de luminancia blanca, p. ej., 100 nits), permite comunicar similarmente las muchas variantes de HDR que pueden tener o generar diversas aplicaciones primarias, y de una manera sencilla. Cuando solo hay que trabajar con un único elemento, en una única aplicación, tal como optimizar un solo video en su

lienzo predeterminado para visualizar, el manejo y procesamiento de HDR no es tan difícil, pero si se usan diversos elementos, aplicaciones y/o lienzos, las cosas pueden volverse rápidamente inmanejables si no se manejan de manera coherente y adecuada. Entonces, cualquier nueva aplicación puede basarse fácilmente en este mecanismo y, después, separar sus propios requisitos de rango de luminancia y formularlos de manera común de acuerdo con el nuevo paradigma. Puede observarse que va más allá de simplemente almacenar datos en la memoria (ya sea después del mapeo de luminancia o no; y con o sin metadatos como, p. ej., una función de procesamiento), ya que el enfoque para el manejo de HDR en el computador puede **coordinar** con una aplicación **secundaria** qué colores estarán finalmente listos para la visualización final (lo cual es muy útil en mezclas de diversos elementos HDR). Asimismo, diversos software en un flujo de manejo de gestión de datos pueden aplicar este mismo método, p. ej., cada uno añadiendo o modificando parte de un conjunto total de elementos HDR locales en una imagen que se va a mostrar.

El software secundario (que puede ser una aplicación para un usuario, o algún software que se ejecuta en segundo plano en el sistema de ventanas) puede ser, p. ej., otro navegador web abierto simultáneamente en otra página, una ventana de un archivo de MS Word o un juego que se mantiene en pausa, etc. También puede ser, si solo

existe una aplicación de usuario que solicita las técnicas de esta solicitud de patente, el sistema operativo. Por lo tanto, p. ej., esta primera aplicación podría coordinarse con el sistema operativo para determinar cómo gestionar los colores de sus elementos y/o cómo no hacerlo. Por computador entendemos cualquier plataforma genérica capaz de ejecutar muchas aplicaciones de software diferentes, incluso simultáneamente (es decir, ocupando su propio espacio en un lienzo visual total, o su búfer de memoria, y en última instancia en una o más pantallas de visualización). Esto implica que típicamente tales computadores tendrían un sistema operativo para coordinar todas las aplicaciones con las funcionalidades de hardware, y por otro lado, el sistema operativo tendría un servidor de ventanas, el cual coordina dónde reside la retroalimentación visual de todas las aplicaciones en la pantalla, y de qué manera. Un computador portátil, como un teléfono móvil, dividirá el espacio de píxeles de manera diferente (y, en cuanto a elementos de la UI simplemente añadirá una barra de estado y una barra de navegación) que, p. ej., un computador de escritorio conectado a una gran pantalla 4K, o incluso un entorno de trabajo multipantalla, pero en lo que respecta a las presentes modalidades técnicas, los principios son los mismos.

Con diferente rango de brillo nos referimos a que el brillo de los píxeles de uno de los elementos

visuales se extiende más allá (típicamente en el extremo más brillante) del brillo del otro elemento. Los rangos de brillo típicamente pueden definirse como rangos de todos los brillos posibles que pueden ocurrir para una imagen (o parte de ella), p. ej., también en un lienzo (aparte de la cantidad de bits y otros acuerdos técnicos de definición de color, que es un tema aparte, un lienzo puede estar trabajando en un rango de, p. ej., 0-5000 nits, por lo que espera que los elementos tengan luminancias de píxeles que estén en algún punto de ese rango). Puede haber un mapeo ascendente, también conocido como actualización (es decir, extensión de rango), de la entrada a la salida, donde la salida típicamente representa un espacio común, pero usualmente habrá un mapeo descendente (también conocido como degradación). En muchas de las modalidades que se describen a continuación, uno de los rangos será de una gama de colores SDR (de hecho, aunque no es necesario para la implementación de las técnicas, sRGB prueba ser un buen modo de implementación, ya que es una manera estandarizada, capaz y bien comprendida de codificar la gama SDR). Para las representaciones HDR relativas (es decir, códigos para los brillos, también conocidos como lumas, que codifican porcentajes de un brillo de referencia que se entiende universalmente, típicamente mediante la estandarización), una imagen HDR mayor que SDR significa que los píxeles pueden tener valores de luma (o luminancia normalizada)

mayores que 1 (o 100 %), y típicamente algunos de los píxeles de una imagen tendrán valores tan altos. Una imagen HDR primaria con un rango mayor que una imagen HDR secundaria significa que la primera puede tener píxeles, p. ej., de hasta más del 500 % (de blanco lambertiano), mientras que la segunda, p. ej., solo puede llegar hasta el 300 %. Las codificaciones de imágenes HDR absolutas tendrán luminancias exactas asignadas a los píxeles (es decir, las luma se entenderán como códigos de valores de luminancia correspondientes en lugar de brillos porcentuales). Una luminancia de referencia típica se asociará a un elemento visual independiente, como una imagen o un video. Una imagen SDR tiene típicamente como luminancia de referencia SDR (es decir, el valor máximo que mostrará el código de luma, p. ej., 1023) 100 nits (esto no debe confundirse con un valor de mezcla para mapear tal elemento visual SDR a un rango HDR, donde se espera que el contenido SDR sea algo más brillante, y a veces puede usarse una luminancia máxima SDR controlable, p. ej., controlable mediante una función de brillo de la UI para el usuario, pero este valor puede típicamente tomarse como 200 nits). Por lo tanto, el brillo será la nomenclatura general en nuestras formulaciones, que abarca tanto el concepto relativo, como el término luminancia más específico, que se usará si alguna parte de la discusión necesita centrarse en el carácter absoluto de la modalidad.

El triplete del código de colores debe entenderse por la persona experta en la presentación gráfica. Para simplificar la explicación, asumiremos que la codificación de color para la visualización aditiva será simplemente RGB, u otra representación popular YCbCr, en la que Y es el código de luma, que codifica el brillo de la impresión visual de un píxel. Aunque algunas pantallas aditivas multiprimarias pueden usar más de tres componentes de color en la generación del color, tales pantallas o codificaciones nunca se popularizaron, por lo que nos mantendremos dentro del paradigma conocido de que tres valores de componentes de color correctamente equilibrados son suficientes para crear prácticamente cualquier color deseado. Sin embargo, la persona experta sabe que puede haber diversas maneras de representar, y codificar, tales triplete de colores. P. ej., también puede usarse el tono, saturación, valor (HSV, por sus siglas en inglés), y algunas aplicaciones usarán y comunicarán tales valores. Independientemente del código que se use, siempre codifica el brillo de un píxel, es decir, al menos su brillo relativo. Las distintas representaciones pueden transformarse entre sí mediante fórmulas colorimétricas fijas y conocidas. P. ej., la luma Y se relaciona con los componentes RGB no lineales (obtenidos a partir de los lineales como resultado de aplicar una OETF) mediante la ecuación $Y=a*R'+b*G'+c*B'$, donde a, b y c son constantes

conocidas. Dado que los detalles exactos de la representación no son relevantes para comprender las contribuciones técnicas actuales, podemos asumir que un triplete de código de color de entrada YCbCr1 se transformará en un triplete de código de color de salida YCbCr2, y solo centrar nuestras enseñanzas técnicas explicativas en esa variante, ya que la persona experta sabe cómo transformar YCbCr2 en coordenadas de control RGB2 para visualizar. El color de salida será el valor de color correspondiente y correctamente coordinado dentro del rango dinámico de diferente tamaño de la imagen de salida, en comparación con el rango dinámico de la imagen de entrada. P. ej., la selección de la función específica a usar puede centrarse en mantener el tono constante entre las dos imágenes y comprimir el brillo de los píxeles, al menos de los objetos más brillantes de la imagen de entrada, en un rango superior más limitado del rango dinámico de salida, al tiempo que se equilibran las saturaciones. Los detalles exactos sobre qué colores desea presentar una aplicación para cualquier elemento visual, o por otro lado, qué funciones de brillo o remapeo de color aplicar con precisión, serán un detalle de cada aplicación, y nuestra solicitud de patente actual es solo un marco técnico para gestionar todas esas situaciones. Por lo tanto, nos limitaremos a ilustrarlo con algunos ejemplos a partir de los cuales se pueda generalizar el concepto.

Una o más funciones de mapeo deben entenderse en términos de los detalles internos de la modalidad de la codificación de color. P. ej., pueden mapearse tres componentes de entrada RGB aplicando una sola función de manera similar a los tres componentes. Eso se hace frecuentemente, y aunque puede introducir algunos cambios de color, puede resultar suficientemente satisfactorio para algunas aplicaciones, algo que también queremos tener en cuenta en nuestro nuevo enfoque. Otro enfoque consistiría en realizar la transformación de brillo en el componente de luma Y, con una primera función, y usar otra función en los valores Cb y Cr para coordinar la croma con el nuevo brillo. En ese caso, por lo tanto, aplicaríamos realmente dos funciones al triplete de entrada, para obtener el triplete de salida correcto. Algunos enfoques, como nuestro SLHDR-1 estandarizado en ETSI 103433, pueden definir el color de salida después de una sucesión de varias funciones a aplicar a las lumas de píxel, p. ej., primero una regradación gruesa y después un ajuste fino. En cualquier caso, todas las funciones necesarias se comunicarán (p. ej., se pasarán como argumentos de una llamada) al módulo de software receptor, y los detalles de si esto ocurre como unos pocos parámetros que definen de manera única una o más funciones (p. ej., a, b y c que definen la forma de una función cuadrática $a*x*x+b*x+c$, si se sabe o se indica que se comunica una función cuadrática), o como una LUT que especifica el

comportamiento de aplicar una o más funciones al mover los valores de entrada de los componentes de color a los valores de salida de los componentes de color, es de nuevo un detalle de los principios técnicos. En general, el procesamiento del color al menos cambiaría el brillo de los colores de los
5 píxeles, independientemente de si se mantienen o no la saturación y el tono. En general, trataremos (explicaremos) el procesamiento como si hubiera una única función que aplicar a cada píxel, aunque diferentes píxeles podrían
10 tener diferentes funciones asociadas.

Nótese que una o más funciones para reasignar los colores en un elemento visual, es decir, en una región del lienzo, no deben confundirse con más de una función típicamente diferente para diferentes elementos o regiones
15 (incluso dentro de un elemento). Para distinguir entre ellas, las hemos denominado funciones primarias de mapeo (F_Y), funciones secundarias de mapeo (F_{Y2}), etc. Para simplificar la comprensión, el lector puede asumir que en la primera región del lienzo total, es decir, el búfer de
20 pantalla, se aplica una primera función cuadrática con primeras constantes a_1 , b_1 , c_1 (p. ej., determinadas por la aplicación de software primaria), y en una segunda región, digamos, por simplicidad, el fondo de la pantalla alrededor de la ventana de la aplicación de software
25 primario, el software secundario, digamos el sistema

operativo, hará que se aplique allí una segunda función de mapeo cuadrático con segundas constantes a_2 , b_2 , c_2 .

Nótese que, en este enfoque de codificación común, típicamente tendremos un procesamiento de rango dinámico, es decir, un cambio de los colores de los píxeles de la imagen original a los colores de los píxeles correspondientes del “formato de imagen común”, que se comunicará típicamente mediante una o más API (p. ej., basadas en sRGB) entre software coordinado en diferentes capas o secuencias. Por lo tanto, esto no debe confundirse con una aplicación que, p. ej., almacena funciones solo para procesar la imagen *original* o al menos una región de esa imagen original, y que almacena tal función junto con la imagen original. Nótese también que el elemento original, incluso si se trata de gráficos simples, de algún creador del elemento es típicamente un elemento creado óptimamente (con el mejor color o el color definitivo) (p. ej., una gradación de color maestra de 5000 nits del elemento), y no, p. ej., un archivo preliminar que aún necesita optimización, tal como una gradación óptima. Es decir, el creador ya ha tomado sus decisiones con respecto a los colores preferidos, que incluyen los brillos o luminancias de los distintos píxeles de su elemento (al menos de manera rápida, pero aún con su aprobación o la de un autómatas inteligente). Por lo tanto, esto no debe confundirse con, p. ej., algún original que ni siquiera

puede verse de manera estable (buena), pero para el cual puede definirse alguna manera rápida de obtener al menos alguna representación visible del mismo (normalmente ninguna de ellas es una gradación maestra óptima, ni una
5 representación diferente de tal gradación de color final).

Los detalles de formato (IMDAT), o su comunicación precisa mediante codificaciones de llamadas a la API especificables, también son irrelevantes para el principio y comprensibles para una persona experta. Por formato nos
10 referimos a cosas como, normalmente, uno o más de los elementos de empaquetado de datos definitivos, representados mediante conjuntos estructurados de códigos, definiciones de términos de API, etc. Los dos módulos de software pueden acordar comunicarse de acuerdo a un espacio de color
15 predefinido (fijo o preconfigurable, p. ej., al abrir la ventana de la aplicación de software primaria por primera vez). Lo explicaremos con dos ejemplos: sRGB o algún espacio de color HDR previamente acordado. Para este último, típicamente se coordina a) una longitud de palabra y una
20 representación binaria, p. ej., un punto flotante de 16 bits para el componente Y, y el Cb Cr; b) un modelo de color, p. ej., RGB codificado en los tres componentes, o YCbCr codificado en las tres palabras de 16 bits; c) una especificación de la geometría precisa, es decir, unidades y
25 tamaño del espacio de color, que típicamente implicará la selección de colores primarios tales como los primarios EBU

RGB, o los primarios DCI-P3 de gama de colores amplia más saturados, y una cromaticidad de punto blanco tal como la luz del día D65; d) una EOTF que especifica cómo los códigos no lineales en el código de representación para brillos lineales correspondientes de magnitudes de componentes RGB lineales, que puede ser lineal en caso de que comuniquemos, p. ej., componentes de color HDR relativos de punto flotante lineal; y opcionalmente e) para espacios HDR absolutos puede coordinarse información adicional relacionada con el rango de brillo numérico, tal como una luminancia de punto blanco de una pantalla de masterización asociada con el contenido de imagen del elemento visual, es decir, para la cual este contenido se califica específicamente (para que sea óptimo sin necesidad de remapear adicionalmente el color).

Para que quede claro para el lector con menos conocimientos de colorimetría: esta alineación de formato de color entre dos procesos de software solo especifica los colores de una imagen (única) *per se*, es decir, p. ej., una gran gama de colores HDR en representación PQ. Como veremos a continuación, también tendremos **funciones de mapeo de luminancia (o luma)**, que definen cómo esta imagen primaria *per se* puede mapearse a y desde los colores de píxeles de una imagen secundaria, que **representará la imagen primaria, al comunicarse entre procesos de software** (pero cualquier proceso también podría, si fuera necesario, reconstruir la imagen primaria a partir de la secundaria tal como se recibe,

aplicando una o más funciones a los tripletes de código de color de píxeles de la imagen secundaria tal como se recibe). Una ventaja de elegir solo unas pocas opciones seleccionables (quizás siempre usando solo sRGB) de formatos prefijados, es

5 que puede comunicarse esto fácilmente mediante un indicador de variante (p. ej., “color_space”==1 significa que todos los códigos de color de p. ej. 3x10 bits son sRGB; o “color_space”==2 significa que los valores deben reinterpretarse a valores lineales usando el EOTF PQ), y

10 entonces uno puede centrarse en lo que realmente importará y variará significativamente en el presente enfoque y modalidades: la una o más funciones (F_Y) (es decir, sus formas particulares elegidas) para aplicar a p. ej. los colores de píxeles sRGB, para, p. ej., volver a obtener

15 composiciones de lienzo HDR, en última instancia en al menos una de las pantallas conectadas siendo una pantalla HDR.

Si una aplicación desea transmitir colores SDR para un elemento, puede, p. ej., añadir una función de identidad. Posteriormente, un software podría cambiar esta función por

20 alguna otra que determine para el elemento, p. ej., una función de actualización. Los colores SDR pueden comunicarse de manera relativa tradicional, o de manera absoluta con una luminancia máxima, p. ej., la luminancia del punto blanco de la pantalla maestra asociada, que suele tener p. ej. un valor

25 de 100 nits para este tipo de elemento SDR.

Los detalles específicos de cómo controlar exactamente, mediante la señal de control (C_PT) las “funciones de paso directo y dejar los colores en la imagen del lienzo transmitido de la aplicación de software primaria sin modificar” (lo que significa que: el segundo software puede, temporalmente, p. ej., en un búfer interno separado, reconstruir los colores HDR aplicando una o más funciones recibidas F_Y sobre los valores sRGB recibidos para ver qué impresión HDR se pretendía, p. ej., en una representación PQ, de manera que pueda determinar **sus propios colores coordinados para mezclar**, p. ej., una barra superior o una barra de desplazamiento de una ventana que no sea demasiado brillante, pero deberá comunicar hacia fuera los colores de la aplicación de software primaria en su región de lienzo **sin mapear**, ya que cualquier mapeo debería ocurrir más adelante, en la GPU o en la pantalla conectada; algunas modalidades pueden cambiar la función cocomunicada F_Y en una función derivada F_Y*), también son irrelevantes para el principio. En algunas modalidades sencillas de API, puede resultar claro, p. ej., a partir de la redacción de la llamada a la API, que también transmite toda la demás información, como p. ej. “FINALIZED PARTIAL_CANVAS(Im_SDR, F_Y, REG1; COLOR_SPACE)”, aunque otras API podrían considerar beneficioso, p. ej., separar al menos la coordinación del espacio de color de la coordinación de la imagen/textura del lienzo parcial.

El principio es que este comportamiento de paso directo instruido hacia la GPU ocurre en uno o más segundos software (intervinientes, en el camino hacia la GPU), tal como lo solicita la aplicación de software primaria.

5 El software terciario es el software final que se encarga de comunicar toda la información correcta de acuerdo con las modalidades a la GPU, es decir, la imagen en color pixelada completamente compuesta y lista para mostrar en el búfer de salida de escaneo (de nuevo en cualquier formato
10 acordado, sRGB, PQ o punto flotante de 16 bits relativo, etc.), y todos los metadatos, es decir, los definidores del espacio de color, las funciones primarias y secundarias (y si corresponde terciarias, etc.), y las regiones de aplicabilidad (REG1*, que se basa en REG1, pero, p. ej., le
15 falta una esquina porque otra ventana se superpone, REG2). En la práctica, frecuentemente, puede ser parte del sistema operativo, es decir, parte del software secundario, en caso de que ese software secundario sea el sistema operativo y no, p. ej., otra aplicación o software intermedio.

20 El efecto técnico, más allá de las técnicas conocidas de este sistema de control para la visualización de elementos visuales (posiblemente en múltiples ventanas), es que, aunque se controle mediante procesos de software, permite una buena coordinación de las necesidades de color
25 HDR de diversos elementos visuales de diferentes procesos y, por lo tanto, en última instancia, una buena apariencia

visual en una pantalla. Además, el marco actual es muy útil cuando no se muestra la imagen en una sola pantalla, sino en diversas pantallas potencialmente muy diferentes (p. ej., una pantalla SDR antigua y una pantalla HDR nueva).

5 Actualmente, esto es un desastre, y a veces ni siquiera puede moverse un elemento de la primera a la segunda pantalla. Con el enfoque actual, la GPU (una o más) puede controlar todos los elementos como se deseaba originalmente y optimizarlos visualmente de manera coordinada para todas

10 las pantallas (lo que significa, por supuesto, que siempre puede haber alguna diferencia entre las pantallas, pero la GPU podría, p. ej., elegir un modo de optimización de pantalla en el que deje los colores tal como se muestran en una segunda pantalla con mayor luminancia máxima

15 sustancialmente iguales que en una primera pantalla con menor luminancia máxima, al arrastrar al menos un elemento definido de esta manera a través de las pantallas).

Lo explicaremos un poco más para que quede más claro con la **Figura 8**, que explica cómo la codificación de

20 colores HDR basada en funciones de mapeo de luminancia o luma (o las correspondientes en otra representación de color, es decir, p. ej., funciones de mapeo para componentes rojos, verdes y azules), y en particular las funciones de actualización de SDR a HDR que son la inversa de las

25 funciones de degradación de HDR a SDR pueden ayudar en la coordinación liberal de la composición geométrica (p. ej.,

interfaz gráfica de usuario GUI) de varios elementos visuales HDR, y en particular si se habilitan diferentes funciones de mapeo para diferentes regiones, de diferentes procesos de software, p. ej., aplicaciones. El solicitante ha inventado y probado que las imágenes o colores HDR pueden representarse/definirse mediante colores SDR correspondientes (“proxy”) (es decir, los colores de una imagen se definen basados en procesamiento de los colores de otra imagen con un rango dinámico diferente), y las ventajas que se derivan de ello, gracias a un nuevos conocimientos, pueden usarse de otra manera técnica, lo que beneficia específicamente a los sistemas de ventanas para computadores, como los que usan, entre otros, las aplicaciones web. El tipo de elementos que se muestran, desde gráficos hasta renderizados de algunos datos (como, p. ej., un mapa meteorológico, o gráfico de algún proceso físico), pasando por imágenes o videos capturados con cámara o generados por IA, logotipos o texto, etc., puede depender de la aplicación web que el usuario esté usando en ese momento, p. ej., una aplicación bancaria, una aplicación de interacción social, o una superposición visual de datos al caminar por la calle, etc.

La ventana izquierda en el lienzo total de la pantalla de la Figura 8C (que corresponde a los mapeos de brillo en la Figura 8A) muestra, p. ej., una película HDR de YouTube que el usuario está viendo, y ese es un ejemplo

de la aplicación de software primaria en nuestra descripción genérica. Digamos que los colores de esta película se definen por un modelo YCbCr basado en PQ, y que la luminancia de los píxeles puede alcanzar un máximo de 2000 nits. Estamos
5 viendo una escena de puesta de sol sobre el mar, que contiene información visual importante en las regiones más brillantes de la imagen (o sus colores representables), concretamente nubes de colores rojo, naranja y amarillo brillantemente iluminadas. Al mapear a sRGB, esos colores deberán
10 comprimirse en un rango superior relativamente estrecho (Rupp; los colores más oscuros en el rango inferior Rlow, p. ej., manteniendo valores de nit aproximadamente iguales en la representación SDR y HDR), pero en cualquier caso, hemos demostrado que esto es posible, incluso con una buena
15 capacidad de reconstrucción de las imágenes HDR maestras originales. En particular, el creador de la vista de YouTube (en última instancia, la aplicación que muestra esta vista, pero que puede depender de otros requisitos colorimétricos y datos del proveedor de esta vista, p. ej., funciones
20 asociadas con la película) tiene la posibilidad de definir de acuerdo con sus preferencias, es decir, cómo considera que es mejor para mantener, p. ej., cierto contraste de luminancia y estructura correspondientes en las nubes, una función de mapeo óptima desde, p. ej., una película de
25 YouTube de 2000 nits potencialmente en una ventana

circundante coordinada de 2000 nits con, p. ej., un rectángulo de fondo, etc., a una gradación sRGB de 100 nits.

Si la aplicación de software primaria usa una función para mapear sus colores HDR maestros originales a los colores de la imagen proxy sRGB (es decir, la imagen que se comunicará a otros procesos de software, p. ej., al menos el sistema operativo, como IMDAT), puede usar la función inversa para recrear y, por lo tanto, definir los colores HDR basados en los colores sRGB. Además, otros software podrían usar el mismo principio de manera exactamente igual. Por lo tanto, en el eje horizontal de la Figura 8A se muestran los brillos relativos de la imagen proxy SDR (p. ej., porcentajes lineales, siendo el 100 % el máximo). Supongamos que la función seleccionada para la aplicación de software primario es la función de mapeo primario F_{Y1} . El sistema operativo (como ejemplo típico de software secundario) puede querer añadir sus propios elementos visuales al lienzo final, es decir, al búfer de pantalla, p. ej., las burbujas de la imagen de fondo (o la barra de la ventana encima del video). El sistema operativo puede considerar que está bien trabajar dentro del marco colorimétrico ya seleccionado y comunicado por la aplicación primaria, por lo que puede reusar la función F_{Y1} para vincular los colores HDR según se desee, y la representación del espacio de color común, que aquí, para simplificar la explicación, se elige como sRGB, que también

es una buena opción desde un punto de vista práctico. Puede decidir por un uso moderado del color, y que las burbujas se sitúen, p. ej., en algún punto intermedio del rango inferior para los objetos de brillo medio. Mediante la

5 curva de mapeo primaria F_{Y1} , cualquier luminancia deseada L_{des_GUI} que se vaya a mostrar finalmente (o una luminancia dependiente si se necesita una adaptación de la pantalla adicional, pero al menos esto era un objetivo universalmente establecido) puede convertirse en un brillo

10 sRGB relativo BR_{in} , es decir, en el brillo deseado Y_{desRGB} , o posiblemente en los valores relativos correspondientes de los componentes RGB. El lector entiende que el OS no necesita necesariamente calcular una imagen HDR a partir de la imagen proxy sRGB recibida, ya

15 que en principio solo necesita aplicar las formas de la función, pero en algunos casos esto puede ser deseable, p. ej., si se necesita calcular un histograma de todos los valores de píxeles que se producen realmente, o incluso si se va a realizar un análisis de patrones espaciales para

20 guiar la elección de los colores de la GUI del OS. Entonces, para que quede claro, la coordinación de colores per se es una tecnología, pero ahora las cosas no se presentan como un escenario calibrado muy simple (en el que todo debe definirse, p. ej., en un rango preestablecido

25 de 4000 nits, que la aplicación web no podría saber qué ocurriría en al menos algún computador receptor), sino que

puede construirse una composición dinámica que permita saber qué colores óptimos deseaba el software primario.

En cualquier caso, en este sencillo escenario, el OS comunicará a la GPU la imagen sRGB compuesta (con todos los elementos visuales en sus posiciones correctas), y la única función `F_Y1`, ya que no se definieron otras funciones. Por lo tanto, aunque son posibles escenarios más complejos, a partir de este sencillo ejemplo vemos que, al menos, el segundo triplete de colores definidos por la aplicación de software primaria no se ve alterado por el OS, ni sus valores codificados (p. ej., YCbCr), ni sus funciones que definen los colores HDR correspondientes. Esto sigue siendo ventajoso al añadir colores de fondo, por un lado, y al requerir un procesamiento final de los segundos tripletes de color, delegándolo a la GPU. Cualquier cálculo de una imagen HDR final, p. ej., para mostrarla en una pantalla mediante la GPU, usará este valor `F_Y1` para determinar **todos** los colores de los píxeles HDR (que ahora han sido coordinados por el OS). Nótese que si la región de aplicabilidad de una función, por defecto, corresponde a toda la pantalla, esto puede comunicarse mediante códigos `REG1*` predeterminados, p. ej., “FULL”, o “NULL”, o lo similar.

Lo que también mostramos es que podemos tener una segunda aplicación (mostrada en la imagen compuesta geométricamente en la Figura 8C), que tiene su propio

objetivo de color y su propia región de la pantalla, que es un anuncio publicitario de bebida de cola, cuyos mapeos de brillo se muestran en la Figura 8B. Supongamos, para mayor claridad, que las dos aplicaciones de software se conocen entre sí, en el sentido de que al menos la aplicación “terciaria”, que es la ventana comercial, tiene acceso al contenido de color de la imagen del contenido de color primario de la aplicación de software primaria (es decir, el visor de YouTube), pero los tripletes de código de color terciarios también pueden generarse de manera totalmente independiente en otros escenarios.

La aplicación terciaria puede, con diversos algoritmos codificados internos, inspeccionar los colores en la película, y tratar de superarlos. P. ej., puede analizar que las regiones más brillantes son las nubes iluminadas por el sol, y definir su información textual (“cola”) en torno a esos niveles de luminancia. Puede hacer que las explosiones de los fuegos artificiales sean aún más brillantes.

Ahora vemos que, si define su propia función de mapeo de luminancia o luma comercial F_{Ycc} , cuya forma aún puede diseñar de manera óptima, puede crear cualquier color deseado, y aun así definirlos en el espacio de color común del búfer de lienzo/pantalla final, que en este caso tiene una representación sRGB. Al comunicar por separado las funciones de mapeo de luminancia, puede, p. ej., tener de repente otro elemento visual que pueda definir sus

luminancias máximas de píxeles hasta 3000 nits, en lugar de 2000. También, al optimizar la forma, la aplicación terciaria puede elegir el nivel de brillo que deben tener el texto y los fuegos artificiales, no solo en su representación HDR
5 maestra, sino también en su lienzo parcial sRGB.

Los detalles sobre cómo un software secundario (en este caso, el ejemplo del OS) puede coordinar la luminancia de sus propios píxeles con lo que ya se encuentra en el lienzo de una aplicación de software primaria, pueden ser
10 muy diversos y estar fuera del alcance de esta solicitud de patente. P. ej., la mezcla puede realizarse en un rango que termina en el mayor de los dos rangos, pero el rango de salida de la mezcla también puede ser algo menor que el rango más grande, porque el compositor quiere abarcar un rango
15 coordinado intermedio (p. ej., el 10 % superior de los brillos del rango más grande se mapea en el rango de salida, con una función derivada correspondiente). Asimismo, el lugar donde el software secundario quiere mapear con precisión los colores de los píxeles de sus objetos puede variar, desde
20 determinarlo únicamente en función de la extensión del rango de salida, hasta considerar lo que ya está dentro del rango de la aplicación de software primaria, p. ej., un histograma de las luminancias de esos píxeles, etc. Pero el lector comprende cómo el OS puede enviar los colores de todos los
25 elementos a la GPU como colores finales (que también pueden definirse de acuerdo con alguna definición de color común,

pero simplemente se reusará el mecanismo seleccionado del software primario, p. ej., sRGB), junto con las diversas funciones de mapeo de luminancia (o brillo) aplicables.

Pero además, este enfoque técnico también ofrece
5 varias ventajas para la GPU. Ahora puede centrarse, p. ej., en una buena adaptación de la pantalla a una **imagen ya compuesta**, y, dado que toda la información de la función de mapeo sigue estando presente, también puede adaptarse a diferentes pantallas, incluso si un elemento visual se
10 desplaza parcialmente de una pantalla a otra. Por lo tanto, la imagen está compuesta en cierta manera, pero a la vez no lo está del todo, ya que las funciones siguen transmitiendo información sobre los diversos objetivos de color originales de las diversas aplicaciones de software,
15 que a su vez podrían basarse en los objetivos de creadores aún más antiguos de un subconjunto de los elementos.

Es ventajoso además que el computador (700) tenga al menos una parte del búfer de salida de escaneo lleno con una copia de los segundos tripletes de código de color (RGB2)
20 como se especifica en la primera estructura de datos de imagen (IMDAT), es decir, sin haber sufrido ninguna modificación colorimétrica como una adaptación de rango dinámico y lo similar, y en donde una codificación derivada de una región espacial (REG1*), que se basa en la codificación de una región
25 espacial (REG1), se comunica a la GPU, que indica para qué píxeles son aplicables la una o más funciones de mapeo

primarias (F_Y) respectivamente, las funciones de mapeo derivadas. La codificación derivada de una región espacial REG1* especificará qué píxeles originales de la aplicación de software primaria siguen siendo visibles, si p. ej. 5 elementos de la GUI del OS u otras ventanas los han cubierto parcialmente, etc. Asumimos opacidad total, y las opciones de transparencia más complejas podrían necesitar funcionalidad adicional, tal como la derivación de una función de mapeo derivada F_Y1*. Esta técnica permite una 10 clara separación entre el “dónde se posiciona geométricamente cada cosa”, es decir, las formas de los objetos visuales, y su tratamiento colorimétrico, más o menos preciso.

Si bien también puede optarse (y probablemente se hará en el futuro) por usar espacios de color HDR 15 estandarizados, dado que el HDR tiene algunas dificultades debido a su complejidad, al menos por el momento, usar sRGB para la comunicación del contenido de la imagen (es decir, lo que en términos gráficos frecuentemente se denomina textura) será una buena opción. Nótese que esta 20 elección se determina de acuerdo con los conocimientos adquiridos por el inventor, considerándola adecuada desde el punto de vista de los sistemas informáticos previstos y para los fines previstos.

Además, resulta ventajoso que una señal de 25 control de paso directo (C_PT) provoque que el software secundario cambie a un modo en el que no aplique un mapeo

de luminancia o luma a los segundos tripletes de código de color (RGB2) en la región espacial (REG1). Ya sería beneficioso comunicar las funciones de mapeo a la GPU per se, de manera que, en algunos casos, esta pueda ser la que decida en última instancia qué colores mostrar. Sin embargo, siguiendo el enfoque actual, la GPU también puede compartir o transferir la responsabilidad de la optimización final de la visualización de los diversos elementos en su composición a la propia pantalla (que son dos modos de control: estricto vs. flexible, o sugerido, pero dicha sugerencia aún comunicaría las diversas funciones de mapeo y sus regiones de aplicabilidad REG1*, REG2, ...). Y algunas modalidades podrían incluso preferir realizar algún ajuste a los tripletes de color de píxeles per se (es decir, los valores sRGB), aunque usualmente se busca mantenerse cerca de las versiones creadas por la aplicación de software primaria; sin embargo, es beneficioso si lo que crea la aplicación de software primaria no se modifica, para que tenga su propio (estricto) control sobre cómo se mostrarán finalmente sus propios colores, al menos de manera aproximada. Usualmente, cualquier procesamiento si es necesario se realizará cambiando las funciones (es decir, aumentando su pendiente, p. ej.), ya que esa es otra ventaja en el marco actual (dado que la gradación secundaria, p. ej., los

colores sRGB, normalmente ya será una gradación perfecta para los distintos elementos).

Es más ventajoso si el computador tiene el software secundario dispuesto para crear un conjunto de
5 tripletes de código de color terciario (RGB3), que están en una misma representación del espacio de color que los tripletes de código de color secundario (RGB2), y para crear una o más funciones de mapeo de color secundario (F_Y2), y una codificación de una segunda región espacial (REG2) de
10 aplicabilidad de una o más funciones de mapeo de color secundario (F_Y2), y para comunicar mediante el tercer software los tripletes de código de color terciario (RGB3) a las posiciones del búfer de salida de escaneo según lo indicado por la segunda región espacial (REG2), y para
15 comunicar la una o más funciones de mapeo de color secundario (F_Y2) a la GPU. Así, p. ej., el OS puede decidir, o incluso imponer, que los colores de sus propios recursos, e incluso los elementos del anuncio publicitario de cola, se definan, p. ej., como sRGB, y se escriban en
20 el búfer de salida de escaneo de la GPU como sRGB.

Resulta ventajoso además que el computador tenga una GPU con un formateador de salida (750) dispuesto para leer, de acuerdo con un escaneo, los colores de píxeles finales del búfer de salida de escaneo (511), formatearlos
25 junto con una o más funciones primarias de mapeo de color (F_Y1) y la codificación derivada de una región espacial

(REG1*) en una señal de interfaz de pantalla (DSIG, por sus siglas en inglés), y comunicar esta señal a una o más pantallas conectadas mediante uno o más mecanismos de comunicación de video (790). Esto puede realizarse p. ej., a través de HDMI, y puede enviarse, p. ej., color sRGB, pero ahora no de la manera típica, sino con toda la información que permite a las pantallas seguir realizando al menos un procesamiento adicional, el cual sea correctamente diferenciado para los distintos recursos, provenientes de diversas fuentes de aplicaciones de software.

Después, las pantallas pueden usar esta información para realizar sus propios ajustes finales en la composición de la GUI, en la medida necesaria, lo que puede resultar ventajoso en algunos casos, pero usualmente se prefiere que el control esté en la GUI, y que las pantallas muestren pasivamente lo que se les indica (es decir, que actúen como monitores de referencia, sin realizar ningún embellecimiento interno). Sin embargo, en algunos casos, algunas pantallas pueden ahora decidir por sí mismas, p. ej., cómo se muestran los anuncios publicitarios.

En otros casos, la GPU tiene un circuito de adaptación de pantalla (751) dispuesto para transformar al menos una de las funciones primarias de mapeo más (F_Y) para producir (por sí mismo) funciones optimizadas de visualización (F_Yd), funciones que deben ser usadas por al menos una de **las pantallas conectadas** para transformar los

segundos tripletes de código de color (RGB2) tal como se comunican mediante el uno o más mecanismos de comunicación de video (790) en tripletes de código de color cuaternarios (RGB4) que están optimizados para la pantalla en el sentido

5 de que ningún triplete de código de color cuaternario (RGB4) será un color que no pueda ser mostrado por la pantalla respectiva. Nótese que, por lo tanto, no es necesario que sea la GPU la que realice los cálculos reales del color de los píxeles para generar la imagen adaptada a la pantalla,

10 sino que puede, por su parte (y teniendo en cuenta sus diversos aspectos), generar las funciones para instruir a los televisores cómo realizar la adaptación de la pantalla (y los televisores aún tienen la opción de desviarse de ello, ya que tienen los tripletes de código de color original

15 y las funciones de guía). Existen varias buenas maneras conocidas de derivar tales funciones, cuyos detalles per se están más allá de los aspectos de la presente innovación, pero puede aprenderse un ejemplo de la Figura 10. Por lo tanto, los colores finales de los píxeles son los que se

20 encuentran en el búfer de salida de escaneo de la GPU, independientemente de si se muestran fielmente, es decir, exactamente como se definen (pueden definirse, p. ej., de acuerdo con una EOTF previamente acordada para definir las palabras clave, que pueden estar fijadas previamente, pero

25 que típicamente también se comunican como metadatos de señalización a la una o más pantallas a través, p. ej., del

cable HDMI antes de suministrar las imágenes), o si todavía se someten a un último mapeo de color en la propia pantalla (que puede realizar alguna implementación local de color), a lo que podríamos llamar colores mostrados finalmente.

5 La manera exacta en que mantiene internamente el lienzo total para más de una pantalla, ya sea como una sola entidad o como búferes separados, es también un detalle que, dadas las posibilidades de las presentes modalidades, resulta irrelevante para la comprensión de los aspectos
10 técnicos y los elementos del presente enfoque. Pero lo importante es que, basándose en las funciones cuyas formas (la forma de una función puede definirse, entre otras cosas, midiendo las distintas distancias de sus puntos a alguna línea de referencia, p. ej., el eje horizontal o una
15 diagonal, y la forma puede adaptarse, p. ej., típicamente ampliando o disminuyendo algunas de esas distancias, típicamente sobre un subconjunto de todos los valores de entrada de la función, es decir, su dominio) indican cómo se vería bien una regradación en comparación con cualquier
20 regradación arbitraria que sea peor, las imágenes finales que se van a mostrar pueden optimizarse al menos a las distintas capacidades de las pantallas y, además, dado que el procesamiento (al menos la determinación de la función de mapeo de luminancia adaptada a la pantalla) se realiza
25 en la GPU, de manera que una pantalla con una capacidad de brillo máximo puede coordinarse con otra de diferente

capacidad (p. ej., la pantalla HDR puede ser controlada con los colores de los píxeles de la imagen por la GPU de manera que al menos los objetos LDR en una imagen parezcan de un brillo similar al de la pantalla SDR, etc.). Esto podría
5 realizarse recolorando los brillos de los píxeles en tiempo real en el búfer de salida de escaneo, mientras una ventana se arrastra parcialmente hacia una nueva pantalla.

Además, puede ser bastante beneficioso si en las modalidades el software secundario se organiza además para
10 derivar una o más funciones de mapeo derivadas (F_{Y^*}) basadas en las funciones primarias de mapeo de color (F_{Y1}) y las funciones secundarias de mapeo de color (F_{Y2}). Esto permite tener en cuenta las necesidades de color de los elementos de las distintas aplicaciones de software que se muestran
15 simultáneamente. Como se explicó anteriormente, el escenario básico sería no modificar las funciones de ninguna aplicación de software, ya que estas también definen los colores básicos deseados por esa aplicación, y la aplicación debería saber qué es lo que queda mejor. Si otro software, p. ej. el OS,
20 considera que todo es suficiente para sus propósitos, puede, como se explicó, definir sus colores en correspondencia con la función (o funciones) F_{Y1} y, por lo tanto, comunicarlo a la GPU sin cambios. Si no está satisfecho con la forma de la función (como ocurrió con la aplicación terciaria en el
25 ejemplo de la Figura 8, que requería una función independiente F_{Y2} debido a que F_{Y1} no alcanzaba el nivel

superior de 3000 nits), normalmente debería definir **su propia función, p. ej. F_Y3**, y comunicarla por separado a la GPU con la región REG3 correspondiente.

Sin embargo, puede haber situaciones en las que
5 algún software, p. ej. el OS, potencialmente bajo el control del usuario (quien, p. ej., indica que no quiere que su película sea interrumpida por algún anuncio publicitario), quiera adaptar al menos algunas de las funciones tal como fueron comunicadas originalmente por una aplicación de
10 software, para hacerlas, p. ej., más coordinadas con otras funciones. P. ej., el OS podría elegir que, a pesar de que el anuncio publicitario quiera presentar fuegos artificiales de 3000 nits que distraigan al usuario, se comunique a la GPU la función secundaria derivada F_Y2*, que es una versión
15 escalada de F_Y2, que termina en 2000 nits, o tal vez incluso solo 1500 o 1000 nits (o incluso simplemente el SDR antiguo que termina en 100 nits, es decir, una transformación de identidad para los colores de píxeles sRGB de REG2).

También puede ser ventajoso incorporar las
20 diversas implementaciones posibles en métodos, que pueden ser realizados, p. ej., por módulos o componentes de software, o aplicaciones y lo similar (y el suministro de diversas maneras de codificar tales métodos):

Un método que se ejecuta en un computador (700)
25 que ejecuta al menos una aplicación de software primaria (701), en la cual la aplicación de software primaria se

dispone para presentar para visualizar elementos visuales de diferentes rangos de brillo, en donde la aplicación de software primaria presenta un primer elemento visual (702) que comprende píxeles tienen brillos que se encuentran dentro de un primer rango de brillo (DR_H);

en donde la aplicación de software primario incluye un mapeo de luma (704) dispuesto para mapear un primer triplete de código de color (RGB1) que codifica un primer brillo de un píxel (7201) a un segundo triplete de código de color correspondiente (RGB2) que codifica un segundo brillo para dicho píxel, aplicando una o más funciones de mapeo primarias (F_Y) al primer triplete de código de color que produce el segundo triplete de código de color, en donde el segundo brillo es un brillo que se encuentra en un segundo rango de brillo (DR_L) de tamaño diferente al del primer rango de brillo, y en donde una o más funciones de mapeo primarias son reversibles (es decir, pueden calcularse los valores de color de salida a partir de los valores de entrada, pero también a la inversa (re)calcular los valores de entrada a partir de esos valores de salida);

en donde la aplicación de software primaria comprende un formato (707) dispuesto para componer segundos tripletes de código de color (RGB2) a partir del mapeo de luma en una primera estructura de datos de imagen (IMDAT) para comunicarse con un software secundario (720), y comunicar al software secundario (720) la primera estructura de datos

de imagen (IMDAT) junto con una codificación de una o más funciones de mapeo primarias (F_Y) y una codificación de una región espacial (REG1) de aplicabilidad de la una o más funciones de mapeo primarias (F_Y);

5 en donde la aplicación de software primaria comprende un controlador primario (710) dispuesto para emitir una señal de control de paso directo (C_PT) que instruye a un controlador secundario (722) en el software secundario (720) que la una o más funciones de mapeo primarias (F_Y), o una o

10 más funciones de mapeo derivadas (F_Y*) que se basan en la una o más funciones de mapeo primarias (F_Y), deben pasarse a un software terciario (740), el cual controla el suministro de colores de píxeles finales para un búfer de salida de escaneo (511) de una unidad de procesamiento de gráficos GPU

15 (510), y en donde el software terciario se dispone para pasar la una o más funciones de mapeo primarias (F_Y) respectivamente o una o más funciones de mapeo derivadas (F_Y*) a la GPU, y en donde al menos algunos colores de píxeles del búfer de salida de escaneo se llenan con copias

20 de los segundos tripletes de código de color (es decir, preferentemente en la misma definición y valores de color).

El método que se ejecuta en un computador (700) en el que al menos una parte del búfer de salida de escaneo se llena con una copia de los segundos tripletes de código de

25 color (RGB2) según lo especificado en la primera estructura de datos de imagen (IMDAT), y en donde una codificación

derivada de una región espacial (REG1*), que se basa en la codificación de una región espacial (REG1), se comunica a la GPU, que indica para qué píxeles son aplicables la una o más funciones de mapeo primarias (F_Y) respectivamente, las
5 funciones de mapeo derivadas.

El método que se ejecuta en un computador (700) en el que el segundo triplete de código de color (RGB2) es una codificación de color de rango dinámico estándar, tal como sRGB.

10 Un método para determinar en una aplicación de software los colores de uno o más elementos visuales que comprenden los píxeles que tienen brillos en un rango dinámico alto (DR H), que comprenden:

- mapeo de luma (704) dispuesto para mapear un
15 primer triplete de código de color (RGB1) que codifica un primer brillo de un píxel (7201) a un segundo triplete de código de color correspondiente (RGB2) que codifica un segundo brillo para dicho píxel, aplicando una o más
20 funciones de mapeo primarias (F_Y) al primer triplete de código de color que produce el segundo triplete de código de color, en donde el segundo brillo es un brillo que se encuentra en un segundo rango de brillo (DR_L) de un
25 tamaño diferente al del rango de alto brillo;

- formato (707) dispuesto para componer segundos tripletes de código de color (RGB2) a partir del mapeo de luma en una primera estructura de datos de imagen (IMDAT) para comunicarse a un software secundario (720) que se ejecuta en un computador (700) que puede ser el mismo computador o un computador diferente del que ejecuta la aplicación de software, y comunicar al software secundario (720) la primera estructura de datos de imagen (IMDAT) junto con una codificación de la una o más funciones primarias de mapeo (F_Y) y una codificación de una región espacial (REG1) de aplicabilidad de la una o más funciones primarias de mapeo (F_Y);
- emitir una señal de control de paso directo (C_PT) para indicar a un controlador secundario (722) en el software secundario (720) que la una o más funciones de mapeo primarias (F_Y) deben pasarse a una unidad de procesamiento de gráficos (510) del computador (700).

Este es el comportamiento que tendría una aplicación de software primaria en el sistema informático y el enfoque explicados anteriormente.

Un método aplicado por un software secundario que se ejecuta en un computador (700) para escribir tripletes de código de color secundario (RGB2) en un búfer de salida de escaneo de una GPU (510), que comprende:

- 5 - recibir de una aplicación de software primaria los segundos tripletes de código de color (RGB2) que junto con una o más funciones de mapeo primarias (F_Y) también recibidas definen los primeros tripletes de código de color (RGB1) que definen los
10 colores de alto rango dinámico de los píxeles de un elemento visual;
- recibir una señal de control de paso directo (C_PT) que instruye a un controlador
15 secundario (722) en el software secundario (720) que la una o más funciones de mapeo primarias (F_Y) deben pasarse a la GPU, y que los segundos tripletes de código de color (RGB2) deben escribirse sin modificación en
20 al menos una región derivada (REG1*) de un búfer de salida de escaneo (511) de la GPU, y que una codificación de la región derivada (REG1*) también debe comunicarse a la GPU.

Un método aplicado por un software secundario, en
25 donde el software secundario determina tripletes de código de color terciarios (RGB3) para una región diferente a la

región derivada (REG1*), y comunica los tripletes de código de color terciarios (RGB3) y una o más funciones secundarias de mapeo de color correspondientes (F_Y2) a la GPU y una codificación de una segunda región (REG2) para la cual los
5 píxeles del búfer de salida de escaneo son aplicables, en donde los tripletes de código de color terciarios (RGB3) se definen en un mismo espacio de color que los tripletes de código de color secundarios (RGB2).

Un método en el que el software secundario es un
10 sistema operativo que se ejecuta en el computador (700).

También puede definirse la invención mediante la información codificada correspondiente al método, p. ej., como una API que define cómo llenar toda la información y leerla para usar en la aplicación de software secundario
15 de este novedoso sistema técnico de gestión de HDR.

BREVE DESCRIPCIÓN DE LAS FIGURAS

Este y otros aspectos del método y aparato de
20 acuerdo con la invención resultarán evidentes a partir de y elucidados con referencia a las implementaciones y modalidades descritas de aquí en adelante, y con referencia a las figuras adjuntas, las cuales sirven meramente como ilustraciones específicas no limitantes que ejemplifican
25 los conceptos más generales, y en las cuales las líneas de trazos se usan para indicar que un componente es opcional,

donde los componentes sin líneas de trazos no necesariamente son esenciales. También pueden usarse líneas discontinuas para indicar esos elementos, que se explica que son esenciales, pero están ocultos en el interior de un objeto, o para cosas intangibles tales como, p. ej., selecciones de objetos/regiones.

En las figuras:

La Figura 1 explica esquemáticamente diversas regradaciones del rango dinámico de luminancia, es decir, mapeos entre las luminancias de entrada (típicamente especificadas según el deseo del creador, humano o máquina, del contenido de video o imagen) y las luminancias de salida correspondientes (de los diversos píxeles de los objetos de imagen), de una serie de etapas o imágenes deseadas que pueden aparecer en una cadena de procesamiento de imágenes HDR;

la Figura 2 introduce esquemáticamente (no limitantes) algunos ejemplos típicos de escenarios de uso de la comunicación de video o imagen HDR desde su origen (p. ej., la producción) hasta su uso (típicamente por un consumidor doméstico);

la Figura 3 ilustra esquemáticamente cómo puede funcionar la conversión de brillo y la conversión correspondiente de los cromas de los píxeles, y cómo pueden verse las distribuciones del histograma de brillo de los píxeles de una imagen correspondiente con un rango dinámico

de brillo menor y mayor, en este caso específicamente de luminancia;

la Figura 4 muestra esquemáticamente cómo las representaciones informáticas de imágenes HDR representarían
5 un rango HDR de luminancias de píxeles de objetos de imagen;

la Figura 5 muestra en un alto nivel los conceptos de la presente aplicación, cómo diversas aplicaciones pueden estar tratando con la visualización de colores de píxeles mediante y/o en competencia con todo
10 tipo de otros procesos de software, lo que puede no dar como resultado un comportamiento de visualización bueno o estable, ciertamente cuando varias pantallas diferentes están conectadas o son conectables;

la Figura 6 muestra un ejemplo arquetípico de un
15 escenario de usuario, en el que el usuario interactúa y visualiza varias aplicaciones web compatibles con HDR en diferentes ventanas que ocupan diferentes áreas de una misma pantalla, y que puede beneficiarse del nuevo enfoque y los elementos técnicos actuales;

20 la Figura 7 muestra esquemáticamente, y entremezcladas, algunas posibles modalidades de la presente tecnología, a un nivel superior;

la Figura 8 muestra esquemáticamente qué ventajas tiene el nuevo marco actual, y cómo el marco puede operar en
25 las definiciones de brillo para una visualización definitiva;

la Figura 9 muestra de manera general una aplicación ilustrativa que funciona como un navegador web;

la Figura 10 muestra una primera manera de optimización de la visualización, mediante la cual la GPU
5 puede determinar una curva de mapeo de brillo o luminancia derivada optimizada en base a una curva de mapeo recibida de una aplicación según el marco actual; y

la Figura 11 muestra otra posibilidad para tal optimización de visualización.

DESCRIPCIÓN DETALLADA DE LAS MODALIDADES

En la **Figura 5** mostramos genéricamente (y esquemáticamente al nivel de detalle necesario) un escenario de aplicación típico en el que funcionarían las modalidades de innovación actuales. Como esto se está volviendo cada vez más popular, mostramos, desde el sistema del lado del cliente (500), que refleja como receptor los datos, comandos y lo similar que se originaron en el lado del servidor (no se muestra), una aplicación web (502). Hemos dibujado la parte de la aplicación web con líneas discontinuas, ya que los problemas y las modalidades que se muestran a continuación pueden funcionar haciendo los cambios pertinentes a las aplicaciones nativas (una “aplicación nativa” es un término de los expertos para una aplicación escrita específicamente para, y que se ejecuta en, una plataforma informática local, tal como una PC o un teléfono móvil).

La aplicación web podría ser, p. ej., un sitio web de banca en línea, un juego en línea, un sitio de video a pedido como Netflix, etc. La aplicación web se ejecutará como software típicamente en una unidad central de procesamiento (CPU) 501. P. ej., la composición de una presentación web puede recibirse como código HTML. La aplicación web se basará en gran medida (para la interacción del usuario) en elementos visuales, tales como p. ej. texto o video.

En la **Figura 6** hemos mostrado de nuevo genéricamente (sin querer ser innecesariamente limitantes) lo que el usuario vería en su una o más pantallas de visualización, mostrando un área o lienzo visible total (el búfer de pantalla 601), que contiene dos ventanas que muestran dos de tales aplicaciones. La aplicación web puede necesitar mostrar sus elementos en alto rango dinámico (es decir, mejor calidad visual, en particular un rango de luminancias de píxeles más brillantes que SDR, y luminancias de píxeles correctas/controladas). P. ej., puede mostrar imágenes fijas HDR que se ven mejor que con un JPEG SDR simple para permitir que el cliente explore las nuevas películas disponibles. O puede que quiera presentar material comercial que ya esté en algún formato HDR (o viceversa, un elemento comercial antiguo que aún no está en HDR, es decir, todavía en Rec. 709).

P. ej., una primera aplicación web puede presentar su contenido en una primera ventana 610. El contenido de esta ventana puede ser, por un lado, un primer texto plano 611 (es decir, colores gráficos para texto estándar, que se representa bien en SDR, p. ej., los colores negro para el texto sobre un fondo “marfil”). Por otro lado, esta página web, que se compone en la primera ventana para el espectador, también puede presentar un video HDR 612 (p. ej., un video transmitido en tiempo real desde una ubicación de archivo en otro servidor, codificado en HLG, y suponiendo que 1000

nits sería un buen nivel para presentar los colores de píxeles de video más brillantes). Esta ventana podría estar una al lado de la otra en un lienzo total del servidor de ventanas, al que aquí llamaremos “búfer de pantalla” (para 5 distinguirlo de los lienzos locales que puedan tener las ventanas). El lienzo es una nomenclatura del campo técnico, que usaremos en este texto para denotar alguna área geométrica, de un conjunto de N horizontal por M píxeles verticales (o similarmente un objeto general como un óvalo), 10 en el que podemos especificar (“escribir” o “dibujar”) algunos píxeles finales, de acuerdo con una representación de color de píxel (p. ej., 3x8 bit R, G, B). Incluso si solo hubiera una ventana de este tipo en el búfer de pantalla, ya podría proyectarse parcialmente en una primera pantalla 15 520 y parcialmente en una segunda pantalla 522, p. ej., al arrastrar esa ventana a través de las pantallas. Si el búfer de pantalla representara los píxeles en un tipo de representación de color, p. ej., SDR, entonces una de las preguntas desafiantes sería cómo la segunda pantalla, que 20 es una pantalla HDR, que espera representaciones de color de píxeles YCbCr definidas por PQ, y puede mostrar colores de píxeles tan brillantes como su capacidad de luminancia máxima ML_D2 de la segunda pantalla, que es de 1000 nits (en comparación con la capacidad de luminancia máxima ML_D1 de 25 la primera pantalla, que es de 100 nits), mostraría su parte de la ventana. El usuario podría encontrar, como mínimo,

extraño o molesto que el fondo blanco del texto pase
repentinamente de 100 nits a 1000 nits. Tenga en cuenta que
también habrá widgets de interfaz gráfica de usuario (GUI),
tal como la barra superior de la primera ventana 613 y la
5 barra de desplazamiento de la primera ventana 614. El
sistema operativo (SO, 504) puede gestionar estos problemas
(aunque a veces también espera que las aplicaciones
gestionen al menos parte de los widgets de la GUI, p. ej.,
ocultando los botones cuando la aplicación está en pantalla
10 completa, y volviéndolos a mostrar al realizar una acción
tal como hacer clic o pasar el cursor por la parte inferior
de la pantalla, o mostrando los widgets parcialmente
transparentes, etc.). Usualmente, al menos por el momento,
los gráficos de estos widgets pueden estar exclusivamente
15 en SDR, pero esto podría cambiar en un futuro próximo.

En la segunda ventana 620 mostramos otra
aplicación web. Las dos aplicaciones web pueden no conocerse
entre sí (y, por tanto, no poder coordinar los colores o sus
niveles de brillo), pero el computador local sí tendrá ese
20 conocimiento (p. ej., la aplicación de servidor de ventanas
del sistema operativo). La segunda ventana, aparte de su
barra superior 623 y su barra de desplazamiento 624, puede,
p. ej., comprender gráficos 3D HDR 622 (por ejemplo, fuegos
artificiales generados por algún modelo físico, con colores
25 muy brillantes, que iluminan un anuncio publicitario) y, en
otra área, un segundo texto 621, que ahora puede, p. ej.,

comprender colores de texto HDR muy brillantes (p. ej.,
definidos en la escala del cuantificador perceptual, por
ejemplo, valores no lineales R_PQ, G_PQ, B_PQ). Las diversas
partes de la pantalla (ventanas, o partes de ventanas,
5 fondo) pueden corresponder a áreas y sus conjuntos de
píxeles, p. ej., un área que comprende exclusivamente todos
los píxeles relacionados con la primera ventana y ninguno
del fondo de la pantalla (mostrada ligeramente más grande
para que sea visible).

10 Actualmente, las aplicaciones acceden a la
funcionalidad de otros programas (es decir, cálculos u
otras acciones que pueden realizar otros procesos de
software) mediante las especificaciones de la interfaz de
programación de aplicaciones (**API**). Esto oculta los
15 detalles más complejos y específicos de, p. ej., otro
software o un componente de hardware como una GPU (como
las instrucciones de código máquina específicas que deben
ejecutarse). Esto funciona porque hay universalidad en las
tareas. P. ej., al usuario final, o a cualquier aplicación
20 de un nivel superior, no le importa cómo una GPU consigue
dibujar exactamente una línea alrededor de una barra de
desplazamiento, sino simplemente que lo haga. También para
los colores de píxeles HDR, aunque puede haber muchos
tipos, al menos la forma en que, p. ej., se define y se
25 crea un color amarillo pastel particular de 750 nits (para
la visualización definitiva) en un píxel determinado, lo

importante es que se lleve a cabo esa acción básica. Lo mismo ocurre con una tubería de renderizado, que usualmente consiste en, p. ej., definir triángulos, colocar estos triángulos en las coordenadas de píxeles correctas, 5 interpolar los colores dentro del triángulo, etc. (y no importa si esto lo hace un solo procesador, incluso al ejecutarse en la CPU, o 20 núcleos de procesamiento paralelo en la GPU, ni dónde y cómo almacenan en caché ciertos datos, etc.).

10 Al menos en la era SDR esto no importaba, porque SDR era justo lo que era, una manera simple y universal de representar todos los colores entre el blanco y el negro y proyectados alrededor de ellos. Y un cierto azul era un cierto azul, y por lo tanto el azul del widget de la GUI 15 se podía mezclar (componer) fácilmente con el azul de la película o de la imagen fija, etc.

P. ej., la aplicación web puede especificar algunas intenciones con respecto a los colores que se deben usar (p. ej., para texto sin importancia, texto importante y texto 20 sobre el que puede pasarse el cursor), p. ej., usando hojas de estilo en cascada (CSS, por sus siglas en inglés), pero puede depender para el renderizado de todos los colores, en un lienzo, de una aplicación nativa 503. Dado que las aplicaciones pueden desear un diseño bien pensado y 25 consistente, la CSS es una forma de separar el contenido (es decir, p. ej., lo que se dice en el texto de un artículo de

noticias) de la definición y comunicación de su presentación, tal como su paleta de colores. Para la navegación web, esta aplicación nativa sería, p. ej., Edge o Firefox, etc. Por lo tanto, puede invocar alguna funcionalidad estándar mediante una primera API (API1). La aplicación nativa puede realizar cierta preparación o procesamiento del texto coloreado, pero también puede apoyarse en alguna funcionalidad del sistema operativo llamando a una segunda API (API2). Sin embargo, el sistema operativo puede, en lugar de realizar algunos cálculos preparatorios en la CPU y después simplemente escribir los colores de píxeles resultantes en la VRAM de la GPU, confiar también en la GPU para calcular los colores desde el principio (p. ej., calcular los fuegos artificiales usando algún sombreador de computación). En cualquier caso, puede emitir una o más llamadas a funciones mediante una tercera API (API3). De hecho, las API gráficas universales modernas como WebGPU, pueden tener ya definidas en la página web las llamadas para la GPU (cuarta API, API11). Un ejemplo de un comando en WebGPU es `copyExternalImageToTexture(source, destination, copySize)`, que copia el contenido de un lienzo de plataforma o imagen a una textura de destino. Las diversas API también pueden consultar, p. ej., cuál es el formato de color preferido de un lienzo (los colores pueden comprimirse de acuerdo con algún formato para ahorrar en comunicación de bus, VRAM necesaria), etc. El búfer final para los colores de la pantalla que se mostrarán, en la GPU 510, es el llamado

ScanOut Buffer 511. Desde allí, los píxeles según sea necesario para comunicarse a la pantalla, p. ej., a través de HDMI, deben escanearse y formatearse en el formato necesario (sincronización, señalización, etc.).

5 Un primer problema es que, aunque uno pueda querer confiar en otro software (de capa) para, p. ej., posicionar y dibujar una ventana, o capturar un evento del ratón, la conversión de rango dinámico es demasiado compleja, y crítica en cuanto a calidad, como para confiar
10 en otros componentes de software para hacer algún mapeo de luminancia o luma probablemente fijo y simplista. El creador de la página web, o el contenido comercial que se ejecuta en ella, puede haber pasado demasiado tiempo haciendo hermosos colores HDR, solo para verlos mapeados
15 con un mapeo de luminancia muy rudimentario, potencialmente incluso recortando información relevante. Si se tratara de una operación de color simple, tal como la corrección de un ligero tono rosado, que tampoco conllevara errores diferenciales muy graves o un aspecto
20 demasiado distinto si no se realiza adecuadamente (o no se realiza en absoluto), se podría confiar en un mecanismo universal para esa tarea, el cual podría ejecutarse en cualquier punto de la cadena. Sin embargo, las imágenes HDR que requieren precisión cromática tienen demasiados
25 detalles como para manejarlas de manera improvisada.

De hecho, hasta hace poco, y como segundo problema, las distintas aplicaciones podían haber definido sus diversos colores HDR (internamente, en su espacio de aplicación), pero los servidores de ventanas (p. ej., el sistema de ventanas de Microsoft) solían asumir que todo se encontraba en el espacio de color SDR estándar **sRGB**. Esto funcionaba bien para componer cualquier tipo de elemento visual SDR. Pero una aplicación que hubiera generado, p. ej., unos fuegos artificiales muy coloridos de 750 nits, necesitaría mapear su luma para convertirla en los colores apagados de unos fuegos artificiales en SDR, y después el servidor de ventanas compondría todo junto en el búfer de pantalla. Por lo tanto, en el mejor de los casos, si la aplicación no dependía ya del sistema operativo, cada una tendría que convertir su hermoso contenido visual HDR a elementos en sRGB, y además no existiría ninguna coordinación con los elementos de otras aplicaciones. Además, en el peor de los casos, podría producirse una cascada de diversos mapeos de luma en las distintas capas de software que se superponen, p. ej., algunos implicando una expansión de SDR a HDR, y todo ello de manera no coordinada. Por consiguiente, no se tendría certeza de cuál sería el resultado final en pantalla, en detrimento de la calidad visual y la experiencia del espectador (p. ej., (partes de) las ventanas podrían verse demasiado oscuras o con contraste insuficiente; si los colores de algunos

píxeles se salen de los límites esperados por cierto software, algunas partes del elemento visual podrían desaparecer al mostrarse, etc.). Recientemente, algunos sistemas de ventanas han introducido la posibilidad de que las aplicaciones puedan comunicar al proceso de composición de ventanas lienzos que puedan contener componentes de color en punto flotante de 16 bits (denominados “medio flotante”) (y, en consecuencia, las luminancias de esos colores). Sin embargo, esto solo permite, comunicar una gran cantidad de colores (per se), pero no garantiza cómo se tratarán en etapas posteriores; p. ej., no asegura qué mapeo de luma se aplicará cuando el sistema operativo realice la composición final de todos los elementos visuales para generar correctamente el búfer de pantalla y, en última instancia, el búfer de salida de escaneo de la GPU. Nótese que la manera exacta en que se gestiona el búfer de pantalla (especialmente cuando intervienen varias pantallas en una vista extendida) en una o más zonas de memoria, mediante una o varias llamadas a la API, etc., no es un aspecto crítico para las modalidades de la presente innovación. P. ej., una pantalla puede actualizarse a una velocidad mayor que la otra, y entonces su parte del lienzo total se leerá con más frecuencia. La innovación consiste en permitir la coordinación óptima de las diversas partes que el usuario llega a ver finalmente, mediante comunicaciones cuidadosas y un manejo armonizado de las representaciones de imagen de

referencia coordinadas (p. ej., sRGB) y las funciones de regradación asociadas para definir elementos HDR, así como el uso coordinado de todo ello cuando, p. ej., se adapta la visualización de la composición final o de parte de esta.

5 En la **Figura 7**, la aplicación de software primaria 701 puede obtener su elemento visual, p. ej., de una memoria 703. La (una o más) función F_Y a aplicar por el mapeador de luma 704 (que está dispuesto para mapear códigos de brillo relativo o luminancias absolutas, 10 típicamente mediante sus códigos, o valores RGB que determinan indirecta pero únicamente los brillos o, respectivamente, las luminancias, o mapeadores HSV, en los que el valor V representa el brillo, etc.) puede ser interna, p. ej., determinada sobre la marcha por la 15 aplicación de software primaria (y almacenada en una memoria interna 705), o (como se indica mediante la flecha punteada que indica el canal de comunicación de datos 706) provenir de una fuente externa. P. ej., si el elemento visual es una película obtenida a través de Internet, la 20 función F_Y puede haber sido determinada ya por el creador de la película, y puede recuperarse de los metadatos de la señal de video de la película como se recibe.

El software secundario siempre tendrá la funcionalidad de control para habilitar el **modo de paso** 25 **directo**, que al menos instruye el paso directo de la(s) función(es) F_Y (o funciones derivadas que conservan la

forma, típicamente algo modificadas, F_Y^*), y típicamente también instruye no tocar, o lo menos posible, los p. ej. colores definidos por sRGB en la imagen del lienzo como se recibe en el IMDAT. Por lo tanto, escribirá tripletes de
5 códigos de color secundarios originales tal como los recibió de la aplicación de software primaria en al menos algunas posiciones correspondientes del búfer de salida de escaneo, p. ej., para aquellas partes de una película que son visibles y no están superpuestas con algún otro elemento visual. Pero
10 algunas modalidades de ese software secundario también pueden tener funcionalidad interna para determinar su propia imagen, y una segunda función de mapeo F_{Y2} , y comunicar eso por separado (también en modo de paso directo) a módulos de software adicionales, o tener un compositor para pasar una
15 imagen mezclada de la imagen sRGB de la aplicación de software primaria y su propio uno o más elementos visuales, típicamente en el espacio de color según las instrucciones de la aplicación de software primaria, es decir, p. ej., siempre el espacio de color sRGB universal. Por lo tanto, el
20 segundo software puede tener su propio algoritmo de mapeo de color 721. (La forma exacta en que se determina la función F_{Y2} puede ocurrir de muchas maneras, y es algo que entiende la persona experta pero que está fuera del alcance de este nuevo enfoque técnico, por lo tanto, simplemente enseñamos
25 que existe otra función además de F_Y). En caso de comunicación por separado a la GPU, esto puede ocurrir en

una representación de datos de imagen secundaria (IMDAT2), pero frecuentemente la GPU recibirá la unión ya compuesta de los dos elementos visuales y sus colores de píxel (IMDAT U IMDAT2), pero frecuentemente con diferentes funciones de mapeo asociadas para las partes de esta matriz de color.

Las modalidades avanzadas del segundo software pueden tener un algoritmo de ajuste de función 725, para determinar la función ajustada F_Y^* basada en F_Y para cualquiera de los elementos visuales (como, p. ej., en el ejemplo de mitigación de brillo comercial explicado con la ayuda de la Figura 8, u otros escenarios como mezclas semitransparentes avanzadas de diversos elementos, o efectos visuales más dinámicos). La caja punteada en la GPU, que es el circuito de adaptación de visualización (751), estará típicamente presente en las GPU preparadas para el futuro, pero en varios escenarios su adaptación de visualización puede estar activada, en cuyo caso pasará una función de mapeo derivada óptima F_{Yd} al formateador de visualización 750 (y frecuentemente también regredará los colores de píxel en el búfer de salida de escaneo correspondientemente, al menos cuando suministra, a través del mecanismo de comunicación de señal de video 790 (p. ej., un cable o conexión inalámbrica), imágenes completamente preoptimizadas). Los detalles del formateador de visualización 750 también dependerán de qué definición de imagen o video se envía a través de qué mecanismo de

comunicación de imagen (debido a la formulación del estándar, HDMI puede permitir diferentes opciones técnicas con respecto a tales aspectos como compresión, ancho de banda, longitud de palabra, espacios de color soportados en 5 códigos de señalización preacordados, etc., distintos a, p. ej., Wifi, por lo que el formateador puede elegir otra opción para formatear la información de la imagen), pero la persona experta entiende esos detalles secundarios. En el otro escenario, donde la adaptación de visualización está 10 desactivada en la GPU, simplemente puede usarse la función original F_Y , y en particular posiblemente comunicarse hacia una pantalla. La señal de imagen típica DSIG que sale de la GPU hacia el mecanismo de comunicación de señal de video 790 comprenderá algunos planos de color de imagen (IM_{So} , p. 15 ej., codificados como sRGB, o como componentes PQ YCbCr o $R'G'B'$), y al menos una función F_Y , si es que una función va a ser aplicada por una pantalla receptora. Frecuentemente, la situación ya habrá sido preoptimizada con respecto a las composiciones, por lo que las pantallas no 20 necesitan recibir varias funciones locales y sus regiones de aplicación correspondientes, pero algunas modalidades de la GPU pueden comunicar esas también, p. ej., para optimizar diferentes ventanas en pantallas, para tales pantallas que serían capaces de regadar los diversos elementos visuales 25 de manera diferente. En general, diversos elementos pueden codificarse, almacenarse, etc. por separado, o al menos

algunos de ellos pueden codificarse juntos, tanto hacia la GPU por la tercera aplicación de software como potencialmente incluso (típicamente en algún grado menor) a la una o más pantallas a través del mecanismo de comunicación de imagen disponible, es decir, Wifi, o 5G, etc.

La aplicación de software primaria también comprende un formateo (707), o formateador, es decir, típicamente algún subalgoritmo o lo similar, que está dispuesto para emitir la diversa información del uno o más elementos visuales de acuerdo con la presente innovación como se explicó, p. ej., esto puede ser alguna señal o llamada de procedimiento predefinida, tal como “HDR-type” {IMDAT, F_Y, REG1}.

El otro elemento técnico nuevo importante del software primario es el controlador primario (710). Este será un mecanismo para coordinar con al menos otra aplicación de software más adelante en lo que respecta al manejo de datos, es decir, recibir la señal o llamada de procedimiento con los datos del elemento HDR, de manera que pueda instruir (además de la mera recepción) cómo manejar este elemento. Típicamente, esto puede ser en detalle, p. ej., también una llamada de procedimiento o señal de instrucción, como p. ej. PASS{Asset_xyz}, en donde Asset_xyz puede ser algún nombre o identificador del elemento, si la llamada de ese elemento se ha comunicado por separado, o se comunicará más tarde, pero en otras

modalidades ambas señales para el software secundario pueden combinarse. En la aplicación de software secundaria 720 (p. ej., el sistema operativo OS), habrá un controlador secundario correspondiente (722), que recibirá esta señal de instrucción y controlará el resto de su procesamiento, en particular su procesamiento colorimétrico. Típicamente, esto llevará a que los datos del software primario estén sustancialmente sin procesar, y simplemente se pasen a la GPU, es decir, a su búfer de salida de escaneo para la matriz de códigos de color, p. ej., los valores sRGB de los píxeles, y su parte (p. ej., una memoria) para recibir las funciones correspondientes y la codificación de las regiones de aplicabilidad. IMDAT2 contendrá típicamente una mezcla de los colores del elemento primario de la aplicación de software primario y lo que sea que la una o más aplicaciones secundarias hayan añadido (geométricamente) a eso, típicamente todo en la misma codificación coordinada, p. ej., sRGB.

El software terciario 740 puede entonces recibir todos los datos y tener el control último de la comunicación hacia la GPU. Aunque en las modalidades algunos de los datos (especialmente en el caso de que la GPU no necesite procesarlos de ninguna otra manera, sino simplemente transmitirlos por el cable, p. ej., HDMI) pueden comunicarse directamente por el OS, p. ej., a alguna parte de la GPU como su formateador de pantalla, típicamente al

menos los datos de imagen que se comunican al búfer de salida de escaneo de pantalla de la GPU se enviarán mediante el software terciario, de manera que se enviarán a la GPU bajo el control de ese software terciario.

5 La **Figura 9** explica brevemente qué puede suceder en detalle si la primera aplicación de software es una página web basada en HTML que se ejecuta en un navegador web. Todos los datos del lado del servidor (es decir, la definición, p. ej., del portal de Netflix) entran como un
10 archivo HTML 901. Contiene información (p. ej., información de ubicación, un enlace a una fuente de datos, etc.) de los diversos elementos de la página web, p. ej., una película 903 reproduciéndose en línea en la página web, y una región de texto coloreado 902. Esto se convierte primero por la
15 aplicación nativa del navegador en un árbol de modelo de objetos del documento (DOM, por sus siglas en inglés) 920, que contiene los datos separados (analizados sintácticamente) en sus propias estructuras de datos, modelo para el video 923 y modelo para el texto 902. Después
20 se construye un llamado árbol renderObject (950), que instruirá típicamente al sistema operativo sobre cómo dibujar y colorear los diversos píxeles, p. ej., del texto. El objeto de video tendrá un renderObject de video 953, que instruye mediante primeras instrucciones Instr1 que los
25 colores de píxel del video deben convertirse, p. ej., a PQ o sRGB, de acuerdo con el algoritmo x, y, o z. El sistema

operativo puede entonces gestionar estos colores, típicamente en un lienzo asociado a ellos, por sí mismo. En caso de que se necesite una función de mapeo de instancia de video separada F_Yv para esta región, tal comunicación de función puede añadirse como una llamada API dedicada (de cualquier otra manera, p. ej., al final puede comunicarse una función general). Los colores de la página web pueden haberse comunicado mediante el mecanismo de hojas de estilo en cascada. El renderObject de texto 952 puede, al instruir sus segundas instrucciones Instr2, aplicar los principios de CSS, y los colores HDR predefinidos para, p. ej., texto de encabezado vs. texto del cuerpo, etc. CSS usa típicamente una cantidad de pasadas de dibujo para dibujar el color de fondo y los objetos del mismo. Esto puede ocurrir ahora, p. ej., en el búfer de pantalla del sistema operativo, p. ej., como colores sRGB correspondientes, que después, como se explicó anteriormente, se pasa a la GPU. O puede ocurrir en un lienzo intermedio temporal, y después el sistema operativo puede copiarlo más tarde al búfer de pantalla y pasarlo a la GPU. Pueden existir temporalmente diversos búferes intermedios para almacenar los resultados de algoritmos complejos de renderizado y/o coloreado. Los elementos visuales que se pasan en procesos separados pueden en el futuro coordinarse más mediante los enfoques elucidados (manteniendo algo de regradación, a través de

las funciones asociadas comunicadas hasta el último nivel(es) de procesamiento).

Las **Figuras 10 y 11** muestran algunos de los beneficios del **control de visualización** de imágenes mediante este mecanismo. Si volvemos al ejemplo de la Figura 8 del video y los elementos visuales comerciales en sus respectivas ubicaciones de ventana (REG1, REG2), podemos preguntarnos qué debería suceder con un video que alcanza un brillo de 2000 nits para al menos algunos píxeles en ciertos instantes, y un anuncio publicitario, quizás una imagen fija, que desea mostrar píxeles con un brillo de 3000 nits, si el usuario ha conectado una pantalla que solo tiene un brillo de 600 nits.

Para hacerlo más interesante, supongamos que en una aplicación de transmisión de pantalla, la pantalla primaria es un teléfono móvil con una impresionante luminancia máxima de 1500 nits, pero de tamaño pequeño, por lo que el usuario quiere transmitir a un monitor secundario económico (aunque con tecnología de recepción, decodificación y visualización de imágenes HDR integrada) con una luminancia máxima de 100 nits, que es grande y tiene resolución espacial 4K. El componente que comunica directamente el video de salida a las pantallas (p. ej., ambos a través de HDMI en formato PQ, aunque las luminancias definidas por PQ-luma para el monitor SDR solo llegarán hasta 100 nits) será la GPU, y con el presente mecanismo de control puede decidir cómo optimizar para ambas pantallas.

Y además, dado que puede tener dos funciones separadas y diferentes para los dos elementos visuales, puede optimizar ambos visuales de esas regiones para que se muestren de manera óptima, no solo basándose en las capacidades de la pantalla, sino también en otras propiedades, tales como p. ej. los requisitos de regradación codificados en la forma de la función, y otros parámetros potenciales indicados a o disponibles en la GPU, tal como un ajuste de brillo del usuario que podría, p. ej., bajar la curva de mapeo comercial F_Ycc como se explicó anteriormente, o de cualquier manera afectando la forma de esa curva (p. e., atenuando las luminancias de salida del subrango más brillante para valores sRGB brillantes (BR_in) por un factor de 3, pero los medios solo por 1,5, y dejando los más oscuros sin cambios).

La GPU puede aplicar su propia función de mapeo final F_Yd (mostrada como una de las maneras alternativas de comportamiento de la GPU en la Figura 7) según lo determinado por algún algoritmo interno para obtener, a partir de los valores sRGB recibidos, y colocar eso en el búfer de salida de escaneo (p. ej., como valores sRGB finales; nota: en diversas implementaciones reales puede haber realmente un búfer frontal y uno trasero separados, por lo que uno de ellos puede ser reoptimizado mientras el otro se muestra continuamente mediante una señal de video comunicada), y esos colores formarán la imagen para la, p. ej., pantalla interna de 1500 nits del teléfono móvil, o

una pantalla de 600 nits que no recibe ningún píxel más brillante que 600 nits en su imagen comunicada, etc.

Digamos que comunica dos regiones de cobertura de pantalla para las dos pantallas, con una señalización que se
5 deriva de los valores sRGB en el búfer de salida de escaneo, p. ej., a la pantalla interna como una señal PQ, y al monitor como una señal rec. 709. La señal rec. 709 puede estar completamente preoptimizada, por lo que el monitor ya no necesita recibir metadatos de función de mapeo que instruyan
10 o guíen el mapeo de luminancia interno del monitor.

La pantalla HDR (interna en este ejemplo, o externa) también puede recibir una señal de video Rec. 709 de SDR (es decir, para los planos de color de píxeles YCbCr), pero también puede recibir una función de mapeo optimizada
15 de la GPU, p. ej., F_{Yd} . Entonces, con esta función, puede reoptimizar los colores SDR recibidos habitualmente según sus capacidades. Esta variante puede funcionar, p. ej., en una única versión de la pantalla transmitida por wifi y recibida habitualmente, o con una imagen duplicada de la
20 pantalla para mostrar la misma pantalla, al menos geométricamente (no necesariamente exactamente igual en cuanto a color), en dos pantallas. Si el lienzo total del búfer de salida de escaneo para dos monitores lado a lado se define, p. ej., en PQ, los valores para el monitor
25 izquierdo de menor luminancia máxima de pantalla pueden comunicarse por cable en formato PQ con valores de luma más

bajos que para el monitor derecho de mayor capacidad de luminancia máxima (reprocesado por la GPU, aplicando la adaptación de pantalla como se explica a continuación).

Por lo tanto, bajo el presente marco, son
5 posibles diversas opciones, dependiendo del nivel de control que se desee dar a la GPU y de lo que aún pueda esperarse de las pantallas conectadas.

La **Figura 10** da un primer ejemplo de un posible algoritmo para realizar la adaptación de pantalla (también
10 conocida como optimización de pantalla). Independientemente de la forma de comunicación, digamos que tenemos conectada solo una pantalla HDR, y necesitamos suministrarle luminancias de píxeles de imagen optimizadas (p. ej., cocomunicadas por HDMI como final, listas para mostrar
15 directamente lumas PQ). La GPU necesita establecer una nueva función de mapeo F_{Yd} , pero sobre la base de la función de mapeo recibida para ese elemento visual y su área. Esto se debe a que la forma de la función se optimizó para lo que los diversos objetos de imagen, y sus
20 correspondientes subrangos de luminancia respectivamente subrangos de luma, necesitan como una regradación óptima hacia un rango de salida menor o mayor. Si se usa un algoritmo que mueve los puntos de la curva de entrada (F_{Yl}) más cerca de la diagonal, pueden obtenerse puntos que
25 definen el lugar geométrico de la función de salida, es decir, la función de mapeo F_{Yd} derivada de la GPU. Esta

función de mapeo derivada F_{Yd} puede usarse para (en los valores de color sRGB, p. ej., como entrada) calcular la imagen de salida completamente optimizada y adaptada a la pantalla para comunicarla desde la GPU a través del
5 mecanismo de comunicación de video a la pantalla para mostrarse directamente (sin mapeo de color adicional).

Tal primer algoritmo ilustrativo dejará todos los puntos mapeados idénticos donde estaban, y puede trazarse, p. ej., desde el punto de identidad superior
10 simplemente una línea hacia la nueva luminancia máxima necesaria (aquí 1000 nits), y después se obtiene la parte superior de la función F_{Yd} .

La **Figura 11** muestra un ejemplo más avanzado de adaptación de pantalla, empleando un enfoque métrico para
15 la luminancia máxima de salida necesaria, y proyección diagonal a la diagonal. Por supuesto, otros algoritmos también pueden hacerlo, ya que esto es nuevamente un detalle del marco: siempre y cuando se comuniquen algunas buenas imágenes desde la GPU a la pantalla, estén o no
20 coasociadas con funciones de mapeo en metadatos para derivar imágenes regradas en las pantallas.

Aquí mostramos una regradación optimizada para la pantalla, es decir, la derivación de la función derivada F_{Yd} partiendo de luminancias HDR. Podemos mapearlas
25 psicovisualmente de manera uniforme usando una OETF adecuada, y después normalizamos a 1,0 como máximo, dando

las diversas lumas de entrada v_HDR de los píxeles (nota, esta representación del espacio de color también puede ser el formato común en el búfer de salida de escaneo, y las diversas comunicaciones de paso directo, en algunas
5 modalidades si así se desea).

En el eje vertical todas las luminancias SDR posibles (es decir, típicamente por acuerdo estándar hasta 100 nits) también se representan normalizadas. Ahora se colocan versiones escaladas de una métrica, en todos los
10 puntos de la diagonal (que corresponden, mediante proyección vertical descendente a las lumas de entrada, p. ej., la primera luma de entrada $vnH1$, y la segunda luma de entrada $vnH2$, respectivamente). Se rotan esas métricas escaladas para que sean ortogonales a la diagonal de 45 grados (diag),
15 y apuntando hacia arriba. El punto en la diagonal corresponde a la luminancia máxima de la representación común HDR, p. ej., 5000 nits. Todos los puntos en las diagonales escaladas que yacen sobre el lugar geométrico de la función de entrada F_Y1^{-1} (cuya función de mapeo de HDR a
20 LDR sería la inversa de la función de mapeo recibida de la aplicación de software primaria para la actualización a HDR, y pasada a través y comunicada a la GPU, o en otras palabras, sería la función que la aplicación de software primaria usó para convertir sus colores HDR de su elemento visual a los
25 colores SDR sRGB) corresponderán a las respectivas posiciones de 100 nits en las respectivas métricas

escaladas. Entonces, puede determinarse una posición en esas respectivas métricas para, p. ej., una pantalla de 600 nits. Las distancias a los puntos de 5000 nits y 100 nits serán ponderadas por una ecuación estándar prefijada, p. ej.,
5 preferentemente calculando las distancias logarítmicamente. Todos los puntos de 600 nits juntos formarán el lugar geométrico de la función de mapeo de salida adaptada a la pantalla, es decir, la función derivada de la GPU F_{Yd} .

Por supuesto, también pueden usarse otros
10 algoritmos de adaptación de pantalla, siempre que resulte una función, y una imagen adaptada en brillo aplicando esa función a la imagen de entrada. Nótese que el OS y la GPU podrían acordar realizar la adaptación de pantalla bajo el control del OS. Las funciones para mapear los cromas
15 correspondientes a un mapeo de luma o luminancia también pueden determinarse de diversas maneras, pero una manera simple y buena es establecer un multiplicador lineal igual a la luminancia de salida dividida por la luminancia de entrada (o el brillo lineal porcentual de salida dividido
20 por el brillo lineal porcentual de entrada), y entonces, después de convertir al formato de los cromas, usar esto como un corrector multiplicativo (p. ej., si la OETF es una raíz cuadrada, se tomará la raíz cuadrada del multiplicador basado en la división para que actúe como el
25 multiplicador final). Los detalles adicionales de esto están muy lejos del alcance necesario para entender la

presente contribución al arte del control del color para pantallas en computadores y entornos informáticos.

El segundo software puede usar principios similares para obtener sus funciones derivadas.

5 Nótese que las regiones tales como REG1 típicamente se superpondrán estrechamente con el lugar donde se van a presentar los elementos visuales, pero también pueden definirse diferentes regiones, p. ej., dividiendo un elemento visual en dos subregiones, etc.

10 Los componentes algorítmicos descritos en este texto pueden (total o parcialmente) realizarse en la práctica como hardware (p. ej., partes de un circuito integrado de aplicación específica) o como software que se ejecuta en un procesador de señal digital especial, o un procesador
15 genérico, etc. Al menos algunos de los elementos de las diversas modalidades pueden ejecutarse en una CPU fija o configurable, GPU, procesador de señal digital, FPGA, unidad de procesamiento neuronal, circuito integrado de aplicación específica, microcontrolador, SoC, etc. Las imágenes pueden
20 almacenarse temporalmente o a largo plazo en diversas memorias, en la proximidad del (de los) procesador(es) o accesibles remotamente, p. ej., a través de Internet.

 Debe ser comprensible para el experto a partir de nuestra presentación cuáles componentes pueden ser mejoras
25 opcionales y puede realizarse en combinación con otros componentes, y cómo las etapas (opcionales) de los métodos

corresponden a medios respectivos de aparatos y viceversa. Algunas combinaciones se enseñarán dividiendo las enseñanzas generales en enseñanzas parciales con respecto a una o más de las partes. La palabra “aparato” en esta solicitud se usa
5 en su más amplio sentido, específicamente un grupo de medios que permite la realización de un objetivo particular, y puede, por lo tanto, p. ej., ser (una parte de un pequeño circuito) un IC, o un aparato dedicado (tal como un aparato con una pantalla), o parte de un sistema en red, etc. El
10 término “disposición” también se prevé que se use en el más amplio sentido, de manera que puede comprender entre otros un único aparato, una parte de un aparato, una colección de (partes de) aparatos que cooperan, etc.

La denotación de producto de programa de
15 computador debe entenderse para que abarque cualquier realización física de una colección de comandos que posibilitan que un procesador genérico o de propósito especial, después de una serie de etapas de carga (que pueden incluir etapas intermedias de conversión, tales
20 como traducción a un lenguaje intermedio, y al lenguaje final de un procesador) introducir los comandos en el procesador, y ejecutar cualquiera de las funciones características de una invención. En particular, el producto del programa informático puede realizarse como
25 datos en un portador tal como, p. ej., un disco, datos presentes en una memoria, datos que viajan mediante una

conexión de red, ya sea por cable o inalámbrica. Aparte del código de programa, los datos característicos requeridos para el programa también pueden incorporarse como un producto de programa de computador. Algunas de las
5 tecnologías pueden estar englobadas en señales, típicamente señales de control para controlar uno o más comportamientos técnicos de, p- ej., un aparato receptor, tal como un televisor. Algunos circuitos pueden ser reconfigurables y configurados temporalmente para un
10 procesamiento particular por software. Algunas partes de los aparatos pueden adaptarse específicamente para recibir, analizar y/o comprender señales innovadoras.

Algunas de las etapas requeridas para el funcionamiento del método ya pueden estar presentes en la
15 funcionalidad del procesador en lugar de descritas en el producto de programa de computador, tales como las etapas de entrada y salida de datos.

Se debe notar que las modalidades mencionadas anteriormente ilustran en lugar de limitar la invención.
20 Donde el experto puede percatarse fácilmente de una asociación de los ejemplos presentados con otras regiones de las reivindicaciones, por concisión no hemos mencionado todas estas opciones en profundidad. Aparte de las combinaciones de elementos de la invención como se combinan
25 en las reivindicaciones, son posibles otras combinaciones de los elementos. Cualquier combinación de elementos puede

realizarse en la práctica en un único elemento dedicado, o elementos divididos.

Ningún signo de referencia entre paréntesis en la reivindicación se destina a limitar la reivindicación.

- 5 La expresión “que comprende” no excluye la presencia de elementos o aspectos no enumerados en una reivindicación. La palabra “un(o)” o “una” que precede a un elemento no excluye la presencia de una pluralidad de tales elementos, ni la presencia de otros elementos.