

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 901 528**

51 Int. Cl.:

H04N 19/105 (2014.01)
H04N 19/176 (2014.01)
H04N 19/103 (2014.01)
H04N 19/186 (2014.01)
H04N 19/157 (2014.01)
H04N 19/85 (2014.01)
H04N 19/70 (2014.01)
H04N 19/46 (2014.01)
H04N 19/61 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **16.03.2006 PCT/US2006/009417**
- 87 Fecha y número de publicación internacional: **26.10.2006 WO06112996**
- 96 Fecha de presentación y número de la solicitud europea: **16.03.2006 E 06738477 (6)**
- 97 Fecha y número de publicación de la concesión europea: **10.11.2021 EP 1869892**

54 Título: **Decodificación de croma que usa predictores independientemente determinados**

30 Prioridad:

13.04.2005 US 671255 P
20.07.2005 US 700834 P

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
22.03.2022

73 Titular/es:

DOLBY INTERNATIONAL AB
Apollo Building, 3E Herikerbergweg 1-13
1101 CN Amsterdam Zuidoost, NL

72 Inventor/es:

YU, HAOPING

74 Agente/Representante:

LINAGE GONZÁLEZ, Rafael

ES 2 901 528 T3

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

DESCRIPCIÓN

Decodificación de croma que usa predictores independientemente determinados

5 Campo de la invención

La presente invención se relaciona en general con métodos y aparatos de decodificación de video.

Antecedentes de la invención

10 En la actualidad, el formato 4:4:4 de norma H.264 de International Telecommunication Union, Telecommunication Sector (Unión Internacional de Telecomunicación, Sector Telecomunicación) (ITU-T) (de aquí en adelante referida como "norma H.264), solamente codifica uno de los tres canales como luma, con los otros dos canales codificados como croma, usando herramientas menos eficientes. Cuando una entrada a un códec está en el formato 4:4:4, con la resolución completa en cada componente de entrada, la codificación de dos de los tres componentes de entrada con un algoritmo de codificación de croma menos efectivo, da como resultado el uso de más bits en esos dos canales. Este problema en particular, es más notable en las intratramas. Por ejemplo, la norma H.264 que corre en el modo solamente interno es menos eficiente que JPEG2k para la calidad de compresión general a 40 dB (PSNR) y anteriores.

20 En consecuencia, sería deseable y altamente ventajoso tener métodos y aparatos para la codificación y decodificación de video que resuelvan las desventajas antes descritas de la técnica anterior.

25 "Propuesta para los problemas no resueltos en extensiones profesionales II, documento JVT-J018", un documento de entrada a JVT de ISO/IEC MPEG y ITU-T VCEG para la 10ª reunión de JVT, 2003, divulga una predicción intracroma con un predictor común para los casos de luma y croma y, de forma más general, permitir que los casos de croma usen los mismos predictores que se conocen para los casos de luma. También divulga un caso en el que para un video YUV con alta resolución, se usa la misma transformación bilineal como filtro de interpolación de compensación de movimiento común.

30 Sumario de la invención

Estas y otras desventajas e inconvenientes de la técnica anterior son abordadas por la presente invención, que se dirige a métodos y aparatos para la decodificación de video.

35 De acuerdo con un aspecto de la presente invención, se proporciona un aparato para decodificar datos de señal de video para un bloque de imagen de acuerdo con la reivindicación 1.

40 De acuerdo con otro aspecto de la presente invención, se proporciona un método para decodificar datos de señal de video para un bloque de imagen en la reivindicación 5.

Estos y otros aspectos, características y ventajas de la presente invención serán evidentes a partir de la siguiente descripción detallada de las realizaciones ejemplificativas, que se debe leer junto con los dibujos anexos.

45 Breve descripción de los dibujos

La presente invención se comprenderá mejor de acuerdo con las siguientes Figuras ejemplificativas, en las cuales:

50 la figura 1 es un diagrama de bloque que ilustra un aparato de codificación de video de ejemplo en el cual se pueden aplicar los presentes principios;

la figura 2 es un diagrama de bloque que ilustra un aparato de decodificación de video de ejemplo en el cual se pueden aplicar los presentes principios;

55 la figura 3 es un diagrama de flujo que ilustra un proceso de codificación de video de ejemplo con una precodificación, un bloque de transformación de color, de acuerdo con los presentes principios;

la figura 4 es un diagrama de flujo que ilustra un proceso de decodificación de video de ejemplo con posdecodificación, un bloque de transformación de color inversa, de acuerdo con los presentes principios;

60 la figura 5 es un diagrama de bloque que ilustra un modelo simplificado de una transformación de color residual (RCT);

65 las figuras 6A y 6B son esquemas de la PSNR promedio contra la velocidad de bits para ATV solamente interno de acuerdo con los presentes principios;

las figuras 7A y 7B son esquemas de la PSNR promedio contra la velocidad de bits para CT solamente interno de acuerdo con los presentes principios.

5 las figuras 8A y 8B son esquemas de la PSNR promedio contra la velocidad de bits para DT solamente interno de acuerdo con los presentes principios;

las figuras 9A y 9B son esquemas de la PSNR promedio contra la velocidad de bits para MIR_HD solamente interno de acuerdo con los presentes principios;

10 las figuras 10A y 10B son esquemas de la PSNR promedio contra la velocidad de bits para RT solamente interno de acuerdo con los presentes principios;

15 las figuras 11A y 11B son esquemas de la PSNR promedio contra la velocidad de bits para STB_HD solamente interno de acuerdo con los presentes principios;

la figura 12 es una tabla que ilustra la sintaxis de parámetros de secuencia H.264 de acuerdo con los presentes principios;

20 la figura 13 es una tabla que ilustra la sintaxis de datos residuales H.264 de acuerdo con los presentes principios;

la figura 14 es un diagrama de flujo que ilustra un proceso de codificación de video de ejemplo con una precodificación, un bloque de transformación de color de acuerdo con los presentes principios;

25 la figura 15 es un diagrama de flujo que ilustra un proceso de decodificación de video de ejemplo con posdecodificación, un bloque del paso de transformación de color inversa, de acuerdo con los presentes principios;

y la figura 16 es una tabla que ilustra la sintaxis de predicción de un macrobloque H.264 de acuerdo con los presentes principios.

30 Descripción detallada

La presente invención está dirigida a métodos y aparatos para la codificación y decodificación de video de datos de señal de video. Se debe apreciar que mientras la presente invención se describe principalmente con respecto a datos de señal de video muestreados de, por ejemplo, el contenido 4:4:4. Las ventajas de esta realización incluyen una mejora en el rendimiento general de codificación para comprimir el contenido 4:4:4 con respecto a la técnica anterior. En la actualidad, en la norma H.264 existente, solamente uno de los tres canales se codifica como luma y los otros dos se codifican como croma usando herramientas menos eficientes.

40 Además, de acuerdo con los principios de la presente invención, como se configuran en una realización, la transformación de color se realiza como un paso de preprocesamiento. De este modo, de acuerdo con esta realización, una transformación de color residual (RCT) no se realiza dentro del ciclo de compresión. Las ventajas de esta realización incluyen que se proporciona una arquitectura del codificador/decodificador más consistente entre los otros formatos de color.

45 Además, de acuerdo con los principios de la presente invención, como se configuran en una realización, se puede utilizar un conjunto (o subconjunto) de tres (3) predictores espaciales restringidos para los tres componentes. Las ventajas de esta realización incluyen una mejora en el rendimiento de codificación general para comprimir el contenido 4:4:4 con respecto a la técnica anterior.

50 Se debe apreciar que las diferentes realizaciones antes descritas y posteriores en el presente documento se pueden implementar como realizaciones independientes o pueden combinarse en cualquier forma, como lo podrán apreciar los expertos en esta técnica y las técnicas relacionadas. De este modo, por ejemplo, en una primera realización combinada no abarcada por las reivindicaciones, un algoritmo de codificación de luma se usa ventajosamente para codificar los tres canales de componente, se realiza una transformación de color como un paso de preprocesamiento, y se usa un solo predictor para los tres canales de componente. En una segunda realización combinada, un algoritmo de codificación de luma se usa ventajosamente para codificar los tres canales de componente, una transformación de color se realiza como un paso de preprocesamiento, y un conjunto (o subconjunto) de tres (3) predictores espaciales restringidos se puede utilizar para los tres canales de componente. Por supuesto, como se menciona antes, se pueden implementar otras combinaciones de las diferentes realizaciones dadas las enseñanzas de los presentes principios proporcionados en el presente documento, mientras se mantengan dentro del alcance de la presente invención.

65 Todos los ejemplos y el lenguaje condicionado nombrados en el presente documento tienen la intención de ser ilustrativos para ayudar al lector a entender los principios de la invención y los conceptos proporcionados por el inventor para mejorar la técnica, y se deben considerar sin limitación para los ejemplos y condiciones nombrados específicamente.

De este modo, por ejemplo, los expertos en la técnica podrán apreciar que los diagramas de bloque presentados en el presente documento representan vistas conceptuales de la circuitería ilustrativa que incorpora los principios de la invención. De manera similar, se debe apreciar que cualquier gráfico de flujo, diagrama de flujo, diagramas de transición de estado, pseudocódigos y sus similares representan varios procesos que se pueden representar esencialmente en un medio legible por computadora y ejecutarse por una computadora o un procesador, ya sea que la computadora o procesador sea o no mostrado explícitamente.

Las funciones de los diferentes elementos mostrados en las figuras pueden ser provistas usando un hardware dedicado así como con un hardware capaz de ejecutar un software asociado con un software apropiado. Cuando es provisto por un procesador, las funciones pueden ser proporcionadas por un único procesador dedicado, mediante un procesador compartido o por una pluralidad de procesadores individuales, algunos de los cuales pueden ser compartidos. Además, el uso explícito del término "procesador" o "controlador" no se debe considerar como refiriéndose exclusivamente a un hardware con la capacidad de ejecutar software y puede implícitamente incluir, sin limitación, un hardware de procesador de señal digital ("DSP"), una memoria de solamente lectura ("ROM") para almacenar el software, una memoria de acceso aleatorio ("RAM"), y un almacenamiento no volátil.

También se puede incluir otro hardware convencional y/o personalizado. De manera similar, los conmutadores mostrados en las figuras son únicamente conceptuales. Su función se puede llevar a cabo a través del funcionamiento de una lógica de programación, a través de una lógica dedicada, a través de la interacción de un control de programa y una lógica dedicada o incluso de forma manual, cuando la técnica particular seleccionada por el desarrollador es más específica para el contexto.

En las reivindicaciones aquí descritas, cualquier elemento expresado como un medio para realizar una función específica tiene la intención de abarcar cualquier forma para realizar esa función, incluyendo por ejemplo a) una combinación de elementos de circuito que realizan esa función; b) software en cualquier forma, incluyendo, por lo tanto, firmware, microcódigos o sus similares, combinados con la circuitería apropiada para ejecutar el software para realizar la función. La invención, como se define por las reivindicaciones reside en el hecho de que las funcionalidades provistas por los diferentes medios nombrados están combinadas y se pueden alcanzar en la forma en que lo describen las reivindicaciones.

Con referencia a la figura 1, un aparato de codificación de video de ejemplo se indica por lo general, con el número 199 de referencia. El aparato 199 de codificación de video incluye un codificador 100 de video y un módulo 105 de transformación de color de precodificación.

El módulo 105 de transformación de color de precodificación es para realizar el preprocesamiento de color de las señales de video antes de introducir las mismas en el codificador 100 de video. El preprocesamiento de color realizado por el módulo 105 de transformación de color de precodificación y se describe además después en el presente documento. Se debe apreciar que el módulo 105 de transformación de color de precodificación se puede omitir en algunas realizaciones.

Una entrada del módulo 105 de transformación de color de precodificación y una entrada del codificador 100 de video están disponibles como entradas del aparato 199 de codificación de video.

Una salida del módulo 105 de transformación de color de precodificación se conecta en comunicación de señal con la entrada del codificador 100 de video.

La entrada del codificador 100 de video se conecta en comunicación de señal con una entrada de no inversión de una unión sumadora 110. La salida de la unión sumadora 110 se conecta en comunicación de señal con un transformador/cuantificador 120. La salida del transformador/cuantificador 120 se conecta en comunicación de señal con un codificador 140 de entropía. Una salida del codificador 140 de entropía está disponible como una salida del codificador 100 de video y también como una salida del aparato 199 de codificación de video.

La salida del transformador/cuantificador 120 también se conecta en comunicación de señal con un transformador/cuantificador inverso 150. Una salida del transformador/cuantificador inverso 150 se conecta en comunicación de señal con una entrada de un filtro 160 de desbloqueo. Una salida del filtro 160 de desbloqueo se conecta en comunicación de señal con los almacenamientos 170 de imagen de referencia. Una primera salida de los almacenamientos 170 de imagen de referencia se conecta en comunicación de señal con una primera entrada de un estimador 180 de predicción espacial y de movimiento. La entrada para el codificador 100 de video también se conecta en comunicación de señal con una segunda entrada del estimador 180 de predicción espacial y de movimiento. La salida del estimador 180 de predicción espacial y de movimiento se conecta en comunicación de señal con una primera entrada de un compensador 190 de predicción espacial y de movimiento. Una segunda salida de los almacenamientos 170 de imagen de referencia se conecta en comunicación de señal con una segunda entrada del compensador 190 de predicción espacial y de movimiento. La salida del compensador 190 de predicción espacial y de movimiento se conecta en comunicación de señal con una entrada de inversión de la unión sumadora 110.

Con referencia a la figura 2, un aparato de decodificación de video de ejemplo se indica por lo general con el número 299 de referencia. El aparato 299 de decodificación de video incluye un decodificador 200 de video y un módulo 293 de transformación de color inversa de posdecodificador.

Una entrada del decodificador 200 de video está disponible como una entrada del aparato 299 de decodificación de video. La entrada del decodificador 200 de video se conecta en comunicación de señal con una entrada del decodificador 210 de entropía. Una primera salida del decodificador 210 de entropía se conecta en comunicación de señal con una entrada de un cuantificador/transformador inverso 220. Una salida del cuantificador/transformador inverso 220 se conecta en comunicación de señal con una primera entrada de una unión sumadora 240.

La salida de la unión sumadora 240 se conecta en comunicación de señal con el filtro 290 de desbloqueo. Una salida del filtro 290 de desbloqueo se conecta en comunicación de señal con los almacenamientos 250 de imagen de referencia. Los almacenamientos 250 de imagen de referencia se conectan en comunicación de señal con una primera entrada del compensador 260 de predicción espacial y de movimiento. Una salida del compensador 260 de predicción espacial y de movimiento se conecta en comunicación de señal con una segunda entrada de la unión sumadora 240. Una segunda salida del decodificador 210 de entropía se conecta en comunicación de señal con una segunda entrada del compensador 260 de entropía. La salida del filtro 290 de desbloqueo está disponible como una salida del aparato de decodificación 299 de video y también como una salida del aparato 299 de decodificación de video.

Además, una salida del módulo 293 de transformación de color inversa de posdecodificación puede estar disponible como la salida del aparato 299 de decodificación de video. En tal caso, la salida del decodificador 200 de video se puede conectar en comunicación de señal con una entrada del módulo 293 de transformación de color inversa de posdecodificación, que es un módulo de posprocesamiento con respecto al decodificador 200 de video. Una salida del módulo 293 de transformación de color inversa de posdecodificación proporciona una señal transformada de color inversa posprocesada con respecto a la salida del decodificador 200 de video.

Ahora se presenta una descripción para la codificación 4:4:4 mejorada de acuerdo con los principios de la presente invención. Una primera realización no reivindicada es una realización combinada en donde se usa un algoritmo de codificación de luma para todos los componentes de color, se usa el mismo modo de predicción espacial para todos los componentes de color, y se omite la transformación de color residual (RCT) desde dentro del ciclo de compresión. Los resultados de prueba para esta realización combinada también se proporcionan. Después, se describe una segunda realización combinada, en donde se usa el algoritmo de codificación de luma para todos los componentes de color, se usa un conjunto (o subconjunto) de predictores espaciales restringidos para todos los componentes de color (en lugar de un único modo de predicción espacial) y se omite la transformación de color residual desde dentro del ciclo de compresión. De este modo, la diferencia entre la primera y la segunda realización combinadas es el uso de un solo modo de predicción espacial para todos los componentes de color en la primera realización combinada contra el uso de un conjunto (o subconjunto) de predictores espaciales restringidos para todos los componentes de color, en la segunda realización combinada. Por supuesto, como se menciona antes, las realizaciones descritas en el presente documento se pueden implementar como realizaciones independientes o se pueden combinar de cualquier manera, como lo podrán apreciar los expertos en esta técnica y las técnicas relacionadas. Se debe apreciar que dadas las enseñanzas de los presentes principios descritos en el presente documento, estas y otras variaciones, implementaciones y combinaciones de las realizaciones de la presente invención serán evidentes para los expertos en esta técnica y las técnicas relacionadas, mientras se mantengan dentro del alcance de la presente invención, que se define en las reivindicaciones adjuntas.

Con referencia a la figura 3, un proceso de codificación de video de ejemplo con un bloque de transformación de color de precodificación se indica por lo general con los números 300 y 301 de referencia, respectivamente.

Se debe apreciar que el bloque 301 de transformación de color de precodificación incluye los bloques 306, 308, y 310. Además, se debe apreciar que el bloque 301 de transformación de color de precodificación es opcional y, por tanto, se puede omitir en algunas realizaciones de la presente invención.

El bloque 301 de transformación de color de precodificación incluye un bloque 306 de límite de ciclo que empieza un ciclo para cada bloque en una imagen y pasa el control a un bloque 308 de función. El bloque 308 de función realiza el preprocesamiento de color de los datos de señal de video del bloque de imagen actual y pasa el control a un bloque 310 de límite de ciclo. El bloque 310 de límite de ciclo termina el ciclo. Además, el bloque 310 de límite de ciclo pasa el control a un bloque 312 de límite de ciclo, éste último siendo incluido en el proceso 300 de codificación de video.

El bloque 312 de límite de ciclo empieza un ciclo para cada bloque en la imagen, y pasa el control a un bloque 315 de función. El bloque 315 de función forma una predicción espacial o compensada por movimiento del bloque de imagen actual usando un predictor común para cada componente de color del bloque de imagen actual y pasa el control a un bloque 320 de función. El bloque 320 de función sustrae la predicción espacial o compensada por movimiento del bloque de imagen actual para formar un residuo de predicción, y pasa el control a un bloque 330 de

función. El bloque 330 de función transforma y cuantifica el residuo de predicción y pasa el control a un bloque 335 de función. El bloque 335 de función transforma y cuantifica de forma inversa el residuo de predicción para formar un residuo de predicción codificado y pasa el control a un bloque 345 de función. El bloque 345 de función añade el residuo codificado a la predicción para formar un bloque de imagen codificada y pasa el control a un bloque 350 de fin de ciclo. El bloque 350 de fin de ciclo termina el ciclo y pasa el control a un bloque 355 de fin.

Con referencia a la figura 4, un proceso de decodificación de video de ejemplo con un bloque de transformación de color inversa de posdecodificación se indica por lo general con los números 400 y 460 de referencia, respectivamente.

Se debe apreciar que el bloque 460 de transformación de color inversa de posdecodificación incluye los bloques 462, 464, 466 y 468.

El proceso 400 de decodificación incluye un bloque 410 de límite de ciclo que empieza un ciclo para un bloque actual en una imagen y pasa el control a un bloque 415 de función. La entropía de bloque 415 de función decodifica el residuo codificado y pasa el control a un bloque 420 de función. El bloque 420 de función transforma y cuantifica de forma inversa el residuo decodificado para formar un residuo codificado y pasa el control a un bloque 430 de función. El bloque 430 de función añade el residuo codificado a la predicción formada de un predictor común para cada componente de color para formar un bloque de imagen codificada y pasa el control a un bloque 435 de límite de ciclo. El bloque 435 de límite de ciclo termina el ciclo y pasa el control a un bloque 440 de fin.

El bloque 435 de límite pasa el control a un bloque 460 de transformación de color inversa de posdecodificación, en particular, el bloque 462 de límite de ciclo incluido en el bloque 460 de transformación de color inversa de posdecodificación. El bloque 462 de límite de ciclo empieza un ciclo para cada bloque en una imagen y pasa el control a un bloque 464 de función. El bloque 464 de función realiza el posprocesamiento de color inversa de los datos de señal de video del bloque de imagen actual y pasa el control a un bloque 466 de límite de ciclo. El bloque 466 de límite de ciclo termina el ciclo y pasa el control a un bloque 468 de fin.

En el formato 4:4:4 de H.264, cada canal de componente tiene una resolución completa. De este modo, de acuerdo con la primera realización combinada antes expuesta, el algoritmo de codificación de luma se usa en cada componente de color para alcanzar la máxima eficiencia de compresión general. En consecuencia, en la realización, para las intratramas, cada componente de color se puede comprimir, por ejemplo, usando esos modos de predicción enumerados en la tabla 8-2, la tabla 8-3, y la tabla 8-4 en la codificación de video avanzada ISO/IEC 14496 10, Tercera Edición (ITU-T Rec. H.264), ISO/IEC JTC1/SC29/WG11 e ITU-T SG16, Q.6, documento N6540, julio de 2004.

Además, en la realización, se usa el mismo modo de predicción espacial para los tres componentes de pixel, para también reducir la complejidad del códec y mejorar el rendimiento. Por ejemplo, el modo de predicción ajustado por los parámetros `prev_intra4x4_pred_mode_flag`, `rem_intra4x4_pred_mode`, `prev_intra8x8_pred_mode_flag` y `rem_intra8x8_pred_mode` para el luma en el encabezado de predicción del macrobloque se puede usar para los tres componentes. Por lo tanto, no se necesitan bits adicionales o elementos de sintaxis adicionales. Para las tramas (predictivas) B y P, los pixeles de referencia en las ubicaciones fraccionadas de pixel pueden calcularse por los métodos de interpolación descritos en la sección 8.4.2.2.1 de la norma H.264 para los tres canales. Los cambios en la sintaxis y semántica detallados para la norma H.264 actual se explican después.

La transformación de color residual (RCT) se añadió al codificador/decodificador en el perfil 4:4:4 alto. Como resultado, la estructura de compresión para el formato 4:4:4 es diferente del actualmente usado en todos los otros perfiles en la norma H.264 para los formatos 4:2:0 y 4:2:2. Esto resulta en cierta complejidad adicional para la implementación. Además, similar a cualquier otra transformación de color, el YCOGC no siempre mejora el rendimiento de compresión general. La efectividad de YCOGC es altamente dependiente del contenido. De esta manera, para mejorar la compresión y la robustez generales, en la realización, la transformación de color se coloca fuera del ciclo de predicción como una parte del bloque de preprocesamiento. Al hacer esto, la selección de una transformación de color óptima para una tarea de compresión específica es un problema operativo y es la mejor respuesta para una secuencia de entrada particular que se pueda encontrarse entre un número de opciones. De acuerdo con una realización, donde todos los tres componentes de color están usando los mismos predictores espaciales para las intratramas y los mismos filtros de interpolación para las tramas (predictivas o intercódificadas) B y P, realizar la transformación de color en los residuos de predicción es idéntico a realizar la transformación de color en las imágenes de origen fuera del códec cuando se ignoran los errores de redondeo/truncado. Esto se explicará más tarde. De este modo, un bloque RCT se elimina de la estructura de codificación para hacer la estructura de codificación consistente entre todos los formatos de color.

Con referencia a la figura 5, el modelo simplificado del RCT se indica por lo general con el número 500 de referencia. El modelo 500 RCT incluye un generador 510 de pixel de referencia, una unión sumadora 520 y un módulo 530 de transformación lineal. Las entradas para el generador 510 de pixel de referencia se configuran para recibir la información de movimiento/borde y los vectores $[X_1]$, $[X_2]$... $[X_n]$. Una salida del generador 510 de pixel de referencia se conecta en comunicación de señal con una entrada de inversión de la unión sumadora 520, que

proporciona el vector $[X_p]$ de predicción al mismo. Una entrada de no inversión de la unión sumadora 520 se configura para recibir el vector $[X_{in}]$ de entrada. Una salida de la unión sumadora 520 se conecta en comunicación de señal con una entrada del módulo 530 de transformación lineal, el cual proporciona el vector $[X_d]$ al mismo. Una salida del módulo 530 de transformación lineal se configura para proporcionar el vector $[Y_d]$.

En el modelo simplificado de RCT 500, la transformación de color representada por una matriz 3×3 $[A]$ (una transformación lineal) se define como sigue:

$$\begin{bmatrix} Y \\ u \\ v \end{bmatrix} = [A] \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

Los $[X_{in}]$, $[X_d]$, $[X_p]$, $[X_1]$, $[X_2] \dots [X_n]$ son vectores 3×1 que representan los píxeles en el dominio RGB. El $[Y_d]$ es un vector 3×1 que representa el resultado de la transformación de color. Por lo tanto,

$$[Y_d] = [A][X_d] = [A][X_m] - [A][X_p] \quad (2)$$

Ya que en la realización, se usan los mismos predictores espaciales y los filtros de interpolación para los tres componentes en un macrobloque de acuerdo con los principios de la presente invención, según se configuran en una realización, el píxel de referencia $[X_p]$ puede expresarse como sigue:

$$[X_p] = [X_1 X_2 X_3 \dots X_n] \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} R_1 & R_2 & R_3 \dots R_n \\ G_1 & G_2 & G_3 \dots G_n \\ B_1 & B_2 & B_3 \dots B_n \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix}, \quad (3)$$

donde un vector (C) $n \times 1$ representa las operaciones lineales involucradas en los predictores espaciales y en los filtros de interpolación definidos en la norma H.264. Aquí, se presume que el píxel de referencia se calcula usando un número total de n píxeles vecinos $[X_1]$, $[X_2] \dots [X_n]$.

Al sustituir $[X_p]$ de la ecuación (3) en la ecuación (2), resulta lo siguiente:

$$[Y_d] = [A][X_{in}] - [A] \left(\begin{bmatrix} R_1 & R_2 & R_3 \dots R_n \\ G_1 & G_2 & G_3 \dots G_n \\ B_1 & B_2 & B_3 \dots B_n \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix} \right). \quad (4)$$

Al ignorar los errores de redondeo/truncado y suponiendo que se selecciona el mismo modo de predicción en el dominio RGB o Y da como resultado lo siguiente:

$$[Y_d] = [A][X_{in}] - \left([A] \begin{bmatrix} R_1 & R_2 & R_3 \dots R_n \\ G_1 & G_2 & G_3 \dots G_n \\ B_1 & B_2 & B_3 \dots B_n \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix} \right) = [Y_{in}] - \begin{bmatrix} Y_1 & Y_2 & Y_3 \dots Y_n \\ u_1 & u_2 & u_3 \dots u_n \\ v_1 & v_2 & v_3 \dots v_n \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix}. \quad (5)$$

Por lo tanto

$$[Y_d] = [Y_{in}] - [Y_1 Y_2 Y_3 \dots Y_n] \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ \vdots \\ C_n \end{bmatrix} \quad (6)$$

De este modo, claramente la ecuación (6) muestra que el uso de YUV como la entrada al codificador/decodificador de acuerdo con los principios de la presente invención como se configura en esta realización, es idéntico a realizar RCT.

También, de acuerdo con los principios de la presente invención como se configura en una realización, un nuevo perfil 4:4:4 se añade a la norma H.264, referido en el presente documento como "perfil 4:4:4 avanzado con profil_idc= 166". Este nuevo profil_idc puede añadirse al encabezado del parámetro de secuencia, y puede ser usado en el encabezado de estrato de macrobloque, así como el encabezado de datos residuales.

Para soportar el uso del algoritmo de luma para codificar los tres componentes de color, se pueden realizar algunos cambios en la sintaxis de datos residuales. Además, los cambios pueden hacerse en la semántica de algunos elementos en el encabezado de macrobloque, en el encabezado de datos residuales y demás. En general, la sintaxis existente para luma en la especificación H.264 quedará sin cambios y será usada para codificar uno de los tres componentes de color. Los cambios tienen compatibilidad con versiones anteriores. Los cambios en sintaxis y semánticas detalladas se describen a continuación.

Ahora se dará una descripción con respecto a los resultados de simulación realizados de acuerdo con los principios de la presente invención, según se configuran en varias realizaciones.

Con referencia a las figuras 6A y 6B, los esquemas de PSNR promedio contra la velocidad de bits para ATV solamente interno se indican por lo general, con los números 600 y 650 de referencia, respectivamente.

Con referencia a las figuras 7A y 7B, los esquemas de PSNR promedio contra la velocidad de bits para CT solamente interno se indican por lo general con los números 700 y 750 de referencia, respectivamente.

Con referencia a las figuras 8A y 8B, los esquemas de PSNR promedio contra la velocidad de bits para DT solamente interno se indican por lo general con los números 800 y 850 de referencia, respectivamente.

Con referencia a las figuras 9A y 9B, los esquemas de PSNR promedio contra la velocidad de bits para MIR_HD solamente interno se indican por lo general con los números 900 y 950 de referencia, respectivamente.

Con referencia a las figuras 10A y 10B, los esquemas de PSNR promedio contra la velocidad de bits para RT solamente interno se indican por lo general con los números 1000 y 1050 de referencia, respectivamente.

Con referencia a las figuras 11A y 11B, los esquemas de PSNR promedio contra la velocidad de bits para STB_HD solamente interno se indican por lo general con los números 1100 y 1150 de referencia, respectivamente.

En particular, las figuras 6A, 7A, 8A, 9A, 10 y 11A ilustran los resultados de prueba para el perfil 4:4:4 avanzado propuesto (indicado y precedido por el término "nuevo") contra los resultados de aproximación correspondientes al mismo. Además, las figuras 6B, 7B, 8B, 9B, 10B, y 11B ilustran los resultados de prueba para el perfil 4:4:4 avanzado propuesto (indicado y precedido por el término "nuevo") contra JPEK2k.

En todas las figuras 6A; 6B hasta 11A, 11B, la PSNR se indica en decibelios (dB) y la velocidad de bits se indica en bits por segundo (bps). ATV, CT, DT, MIR, RT, STB son los nombres de los clips de prueba.

Todas las secuencias de prueba JVT/FRExt descritas en las secuencias de prueba originadas por película JVT-J042, se usaron en las pruebas. Todas ellas son material de película de 10 bits 4:4:4 y cada clip tiene 58 tramas.

Los perfiles 4:4:4 avanzados propuestos se implementaron en el software de referencia JVT JM 9.6. Tanto la solamente interno y la estructura de codificación IBBP se usaron en las pruebas. El parámetro de cuantificación se estableció en 6, 12, 18, 24, 30 y 42 para cada una de las curvas R-D. Se usó la selección de modo optimizado por RD.

El perfil 4:4:4 avanzado propuesto también se comparó con los resultados que se llevaron a cabo al ejecutar el software de referencia con el YUVFormat = 04:0:0) en cada componente individual de entrada. Tres cuentas de bits

comprimidos individuales separadas se añadieron juntas para obtener los bits comprimidos totales para calcular la velocidad de bits comprimida.

Con respecto a JPEG2k, se usó el software KaKadu V2.2.3 en las pruebas. Los resultados de prueba se generaron usando 5 niveles de la descompresión de forma de onda, con el filtro de forma de onda biortogonal de 9/7 de toma. Esto fue solamente una tesela por trama y la optimización por RD para una velocidad de destino dada también se usó.

Todas las mediciones PSNR se llevaron a cabo en el dominio RGB. La PSNR promedio definida como $(\text{PSNR}(\text{rojo}) + \text{PSNR}(\text{verde}) + \text{PSNR}(\text{azul}))/3$ se usa para comparar la calidad de compresión general. Esto es principalmente porque los datos comprimidos JPEG2K se computan usando un algoritmo de control de velocidad desconocida provisto por el software. Para ciertos casos, los valores RGB PSNR están un poco alejados entre sí, especialmente cuando se usa la transformación de color JPEG2k.

La comparación de compresión se realizó como sigue:

- New1: el perfil 4:4:4 avanzado propuesto con un solo modo de predicción.
- New3: el perfil 4:4:4 avanzado propuesto con tres modos de predicción.
- RCT-OFF: entrada RGB con RCT = apagado.
- RCT- ON: entrada RGB con RCT = encendido.
- YCOCG: la conversión RGB a YCOCG se hizo fuera del códec. Entonces el YCOCG convertido se usó como la entrada para el software JVT.
- R + G + B: método propuesto aproximado al comprimir las señales R, G y B por separado.
- Y + CO + CG: método propuesto aproximado al comprimir las señales de Y, CO y CG convertidas de forma separada.
- J2k_RGB: la compresión JPEG2 se hizo en el dominio RGB. Se apagó la transformación de color JPEG2k.
- J2k_YUV: La compresión JPEG2k se hizo en el dominio YUV. Se usó la transformación de color JPEG2k.

De acuerdo con los resultados de prueba, una implementación de acuerdo con los principios de la presente invención se configura en una realización, en general y es muy similar a JPEG2k en términos de eficiencia general de compresión. En algunos casos, es incluso un poco mejor.

Además, una implementación de acuerdo con los principios de la presente invención, según se configuran en una realización, proporciona mayor rendimiento (compresión) que el perfil 4:4:4 alto actual para una calidad sobre 40 dB (PSNR). Específicamente, New1-YCOCG o New3- YCOCG es mejor que YCOCG y RCT-ON, New1-RGB o New3-RGB es mejor que RCT-OFF. A una PSNR igual o mayor que 45 dB (PSNR) la mejora promedio en la PSNR promedio es más de 1,5 dB. En el último ejemplo, la mejora se puede traducir en más de un 25% de ahorro de bits en una PSNR igual a 45 dB.

De acuerdo con los resultados de prueba, parece que las transformaciones de color ayudarán al rendimiento de codificación cuando el contenido está más saturado de color, tal como TP, RT. Esto es, si el color es neutral y menos saturado, la codificación en el dominio RGB puede ser la opción correcta. La observación anterior es independiente de la transformación de color que se use.

Al comparar los resultados de New1-YCOCG o New3-YCOCG y JPEG-2k_YUV, se ha observado que el rendimiento de una transformación de color específica en términos de mejorar la eficiencia de codificación depende mucho del contenido. Una sola transformación de color no es siempre la mejor. Por lo tanto, nuestros datos confirmaron que al tener una transformación de color, tal como RCT, dentro del ciclo de codificación (o decodificación) puede no ser una buena idea. En su lugar, al llevar a cabo la transformación de color, cuando es necesario, fuera del codificador/decodificador puede hacer que el sistema de compresión completo proporcione un rendimiento mejor y más robusto.

Al comparar YCOCG con RCT-ON, los resultados de prueba no muestran ninguna mejora en la eficiencia de codificación de RCT: Además, cabe señalar que al ejecutar el software de referencia con el RCT encendido incrementó mucho el tiempo de codificación. El tiempo de ejecución fue 2,5 veces más largo.

Ahora se dará una descripción con respecto a los cambios en sintaxis y semántica de acuerdo con los principios de la presente invención como se configura en una realización.

Con referencia a la figura 12, una tabla para la sintaxis del parámetro de secuencia H.264 se indica por lo general con el número 1200 de referencia. Los cambios en sintaxis de acuerdo con los principios de la presente invención como se configura en una realización, se indican con el texto en cursiva.

Con referencia a la figura 13, una tabla para la sintaxis de datos residuales se indica por lo general con el número 1300 de referencia. Los cambios/adiciones en la sintaxis de acuerdo con los principios de la presente invención según se configuran en una realización, se indican con texto en cursiva. En la tabla 1300, la sección de luma en el encabezado de datos residuales junto con ciertas modificaciones de texto necesarias se repiten dos veces para dar soporte a luma1 y luma2, respectivamente.

Como se señaló antes, la primera realización combinada se evaluó y se probó al implementar los presentes principios en el software de referencia JVT JM9.6. Los resultados de prueba marcados con New1- RGB o New1- YCOG representan la primera realización combinada. Como se señaló antes, de acuerdo con los principios de la presente realización según se configuran en una realización, un conjunto (o subconjunto) de tres (3) predictores espaciales restringidos se utiliza para los canales de componente (por ejemplo, los formatos RGB, YUV, YCrCb y demás), en lugar de un único modo de predicción espacial. Además, como se señaló antes, esta realización se puede combinar con otras realizaciones descritas en el presente documento, tal como por ejemplo, el uso de solamente el algoritmo de codificación de luma para codificar los tres canales de componente de contenido y/o el uso de la transformación de color como un paso de preprocesamiento.

Ahora se dará una descripción con respecto a la segunda realización combinada antes descrita, la cual involucra el uso de un conjunto (o subconjunto) de tres (3) predictores espaciales restringidos para los componentes de color, el uso de un algoritmo de codificación de luma para codificar los tres componentes de color, y el uso de una transformación de color como un paso de preprocesamiento (es decir, sin RCT dentro del ciclo de compresión). Algunas variaciones de esta realización también serán descritas

Con referencia a la figura 14, un proceso de codificación de video de ejemplo con un paso de transformación de color de precodificación se indican por lo general con los números 1400 y 1401 de referencia, respectivamente.

Se debe apreciar que el bloque 1401 de transformación de color de precodificación incluye los bloques 1406, 1408, y 1410. Además, se debe apreciar que el bloque 1401 de transformación de color de precodificación es opcional y, por lo tanto, se puede omitir en algunas realizaciones de la presente invención.

El bloque 1401 de transformación de color de precodificación incluye un bloque 1406 de límite de ciclo que empieza un ciclo para cada bloque en una imagen, y pasa el control a un bloque 1408 de función. El bloque 1408 de función realiza el preprocesamiento de color de los datos de señal de video del bloque de imagen actual y pasa el control a un bloque 1410 de límite de ciclo. El bloque 1410 de límite de ciclo termina el ciclo. Además, el bloque 1410 de límite de ciclo pasa el control a un bloque 1412 de límite de ciclo, éste último estando incluido en el proceso 1400 de codificación de video.

El bloque 1412 de límite de ciclo empieza un ciclo para cada bloque en la imagen, y pasa el control a un bloque 1415 de función. El bloque 1415 de función forma una predicción espacial o compensada por movimiento del bloque de imagen actual usando un predictor común para cada componente de color del bloque de imagen actual, y pasa el control a un bloque 1420 de función. El bloque 1420 de función sustrae la predicción espacial o compensada por movimiento del bloque de imagen actual para formar un residuo de predicción y pasa el control a un bloque 1430 de función. El bloque 1430 de función transforma y cuantifica el residuo de predicción y pasa el control a un bloque 1435 de función. El bloque 1435 de función transforma y cuantifica de forma inversa el residuo de predicción para formar un residuo de predicción codificado y pasa el control a un bloque 1445 de función. El bloque 1445 de función añade el residuo codificado a la predicción para formar un bloque de imagen codificada y pasa el control a un bloque 1450 de fin de ciclo. El bloque 1450 de fin de ciclo termina el ciclo y pasa el control a un bloque 1455 de fin.

Con referencia a la figura 15, un proceso de decodificación de video de ejemplo con un paso de transformación de color inversa de posdecodificación se indican por lo general con los números 1500 y 1560 de referencia, respectivamente.

Se debe apreciar que el bloque 1560 de transformación de color inversa, de posdecodificación incluye los bloques 1562, 1564, 1566 y 1568.

El proceso 1500 de decodificación incluye un bloque 1510 de límite de ciclo que empieza un ciclo para un bloque actual en una imagen y pasa el control a un bloque 1515 de función. El bloque 1515 de función decodifica con entropía el residuo codificado y pasa el control a un bloque 1520 de función. El bloque 1520 de función transforma y cuantifica de forma inversa el residuo decodificado para formar un residuo codificado y pasa el control a un bloque 1530 de función. El bloque 1530 de función añade el residuo codificado a la predicción formada desde un predictor común para cada componente de color para formar un bloque de imagen codificada y pasa el control a un bloque 1535 de límite de ciclo. El bloque 1535 de límite de ciclo termina el ciclo y pasa el control a un bloque 1540 de fin.

El bloque 1535 de límite de pasa el control al bloque 1560 de transformación de color inversa de posdecodificación, en particular al bloque 1562 de límite de ciclo incluido en el bloque 1560 de transformación de color inversa de posdecodificación. El bloque 1562 de límite de ciclo empieza un ciclo para cada bloque en una imagen y pasa el control a un bloque 1564 de función. El bloque 1564 de función realiza el posprocesamiento de color inversa de los datos de señal de video del bloque de imagen actual y pasa el control a un bloque 1566 de límite de ciclo. El bloque 1566 de límite de ciclo termina el ciclo y pasa el control a un bloque 1568 de fin.

Como se mencionó antes, se divulga un nuevo perfil (perfil_idc=166) para el perfil 4:4:4 avanzado. Este nuevo perfil también se puede usar para la segunda realización combinada, con los cambios de semántica y sintaxis correspondientes según se describe en el presente documento para la segunda realización combinada. Este nuevo `profile_idc` se añade al conjunto de parámetros de secuencia y será usado principalmente en los encabezados posteriores para indicar que el formato de entrada es 4:4:4 y los tres canales de entrada se codifican de forma similar al luma.

Para minimizar los cambios necesarios en la norma H.264, no se describe ningún tipo nuevo de macrobloque para el perfil 4:4:4 avanzado. En su lugar, todos los tipos de macrobloque junto con los parámetros de codificación asociados enumerados en la tabla 7-11, la tabla 7-13 y la tabla 7-14, de la norma H.264 siguen siendo válidos. Para el caso de intramacrobloques, los tres canales de entrada, luma, Cr y Cb se codificarán basándose en `MbPartPredMode` definido en la tabla 7-11 de la norma H.264. Por ejemplo, un macrobloque `intra_4x4` en el perfil 4:4:4 avanzado significa que cada canal de componente de entrada se puede codificar usando los 9 posibles modos de predicción determinados en la tabla 8-2 de la norma H.264. Como referencia, en el perfil 4:4:4 alto, dos de los canales para el macrobloque `intra_4x4` serán tratados como croma y solamente se usará uno de los cuatro posibles modos de intrapredicción en la tabla 8-5 de la norma H.264. Para los macrobloques B y P, los cambios hechos para el perfil 4:4:4 avanzado ocurren en el proceso de interpolación para el cálculo del valor del píxel de referencia en la ubicación fraccionada del píxel. Aquí, el procedimiento descrito en la sección 8.4.2.2.1 de la norma H.264, el proceso de interpolación de muestra de luma, se aplicarán para el luma, Cr y Cb. Otra vez, como referencia, el perfil 4:4:4 actual usa la sección 8.4.2.2.2 del proceso de interpolación de muestra de croma de la norma H.264, para dos de los canales de entrada.

En el caso en el que se selecciona CABAC como el modo de codificación de entropía, dos conjuntos separados de modelos de contexto idénticos a los actualmente definidos para luma serán creados para Cr y Cb. También se pueden actualizar de forma independiente durante el curso de la codificación.

Por último, en la realización, ya que no hay un bloque RCT en el ciclo de codificación, el `ResidueColorTransformFlag` se elimina del conjunto de parámetro de secuencia en el perfil 4:4:4 avanzado.

Hasta este punto, la mayoría de los cambios en sintaxis ocurren en los datos de residuo como se muestra en la figura 13, donde la sintaxis original para el luma se repite dos veces para dar soporte a Cr y Cb en los perfiles 4:4:4 avanzados propuestos.

Con respecto a la tabla de estrato de macrobloque H.264 (no mostrada), los cambios semánticos correspondientes a la sintaxis incluyen lo siguiente.

`coded_block_pattern` (ADD). Cuando `chroma_format_idc` es igual a 3 y `coded_block_pattern` está presente,

`codedBlockPatternChroma` debe establecerse en 0. Además, `codedBlockPatternLuma` especifica, para cada uno de los doce bloques 8x8 luma, Cb y Cr del macrobloque, uno de los siguientes casos: (1) todos los niveles de coeficiente de transformación de los doce bloques de luma 4x4 en los bloques de luma 8x8, Cb 8x8 y Cr 8x8 son iguales a cero; (2) Uno o más de los niveles de coeficiente de transformación de uno o más de los bloques de luma 4x4 en los bloques de luma 8x8, Cb 8x8 y Cr 8x8 serán valorados en no cero.

Ahora se dará una descripción con respecto a la selección de modo de predicción espacial para los intrabloques de acuerdo con la segunda realización combinada (o la única realización relacionada con el uso del conjunto (o subconjunto) de los tres predictores espaciales restringidos).

Para que cada componente de color seleccione su mejor `MbPartPredMode` y el mejor modo de predicción espacial posterior de forma independiente, como en el caso mientras se codifica cada canal de entrada por separado, se pueden añadir algunos tipos de intrabloque en la tabla 7-11 de la norma H.264. Como resultado, se realizarán una gran cantidad de cambios en la norma H.264. En una realización relacionada con la segunda realización combinada, los `mb_types` actuales permanecen sin cambio y se proporciona una solución alternativa. En la realización, los tres canales de entrada están restringidos para ser codificados con el mismo `MbPartPredMode` o tipo de macrobloque. Entonces una pequeña cantidad de nuevos elementos se añaden dentro de la sintaxis de predicción del macrobloque para dar soporte a tres modos de predicción separados. Por lo tanto, cada componente puede elegir teóricamente su mejor modo de predicción espacial, de forma independiente, con el fin de reducir al mínimo el error de predicción para cada canal de componente. Por ejemplo, suponiendo que se selecciona un macrobloque 4x4

como el mb_type, luma, Cr o Cb puede encontrar su propio modo de predicción espacial en la tabla 8-2 en la sección 8.3.1.1 de la norma H.264, tal como por ejemplo, Intra_4x4_Vertical para luma, Intra_4_4_Horizontal para Cr e Intra_4x4_Diagonal_Down_Left para Cb.

- 5 Otro enfoque relacionado con la primera realización combinada descrita antes, es restringir los tres canales de entrada para compartir el mismo modo de predicción. Esto se puede realizar usando la información de predicción que es llevada actualmente por los elementos de sintaxis existentes, tales como prev_intra4x5_pre_mode_flag, rem_intra4x5_pred_mode, pre_intra8x8_pred_mode_flag y rem_intra8x8_pred_mode, en la sintaxis de predicción del macrobloque. Esta opción provocará un cambio menor en la norma H.264 y en cierta pérdida de la eficiencia de codificación también.

Basándose en los resultados de prueba, usar tres modos de predicción puede mejorar el rendimiento de codificación general aproximadamente 0,2 dB sobre la primera realización combinada.

- 15 Con referencia a la figura 16, se indica por lo general una tabla para la sintaxis de predicción de macrobloque H.264 con el número 1700 de referencia. Como referencia, se enumera a continuación la sintaxis de predicción de macrobloque modificada para dar soporte a tres modos de predicción, donde:

prev_intra4x4_pred_mod0 y rem_intra4x4_pred_mod0 son para luma;

- 20 prev_intra4x4_pred_mod1 y rem_intra4x4_pred_mod1 son para Cr;

prev_intra4x4_pred_mod2 y rem_intra4x4_pred_mod2 son para Cb.

- 25 Ahora se dará una descripción con respecto a los resultados de simulación realizados de acuerdo con los presentes principios de la invención, según se configuran en una realización, para la segunda realización combinada.

Todas las secuencias de prueba JVT/FRExt descritas en la secuencia de prueba originada por película, JVT-J042, JVT-J039 (Viper). Todas ellas son materiales de 10 bits y cada clip tiene 58 tramas.

- 30 El algoritmo propuesto se implementó en un software JM9.6 de referencia JVT y el software modificado se usó en las pruebas. Se probaron tanto solamente interno como IBRRBP. Aquí, "Br" significa las imágenes B grabadas. El caso solamente interno se realizó para todas las secuencias con el parámetro de cuantificación igual a 6, 12, 18, 24, 30, 36 y 42. Debido a la gran cantidad de tiempo involucrado en la simulación, la estructura GOP IBRRBP se llevó a cabo para los clips de película con un parámetro de cuantificación igual a 12, 18, 24, 30 y 36. De acuerdo con esta explicación en el AHG 4:4:4, se usaron los siguientes parámetros clave en las pruebas:

SymbolMode = 1
RDOptimization = 1
40 ScalingMatrixPresentFlag = 0
OffsetMatrixPresentFlag = 1
QoffsetMatrixFile = "q_offset.cfg"
AdaptiveRounding = 1
AdaptRndPeriod = 1
45 AdaptRndChroma = 1
AdaptRndWFactorX = 8
SearchRange = 64
UseFME = 1

- 50 Con respecto a JPEG2K, se usó el software KaKadu V2.2.3 en las pruebas. Los resultados de prueba se generaron usando 5 niveles de la descompresión de forma de onda, con el filtro de forma de onda biortogonal de 9/7 de toma. Solo había una tesela por trama y también se usó la RDOptimization para una velocidad de destino dada.

- 55 Las mediciones PSNR se calcularon en primer lugar en el dominio de color original de los contenidos de origen, que es RGB para los clip antes descritos. La PSNR promedio, definida como (PSNR(rojo) + PSNR(verde) + PSNR(azul))/3 se usa para comparar la calidad de compresión general.

La comparación de compresión se realizó como sigue:

- 60 New1: el perfil 4:4:4 avanzado propuesto con un solo modo de predicción.
New3: el perfil 4:4:4 avanzado propuesto con tres modos de predicción.
RCT-OFF: entrada RGB con RCT = apagado.
RCT-ON: entrada RGB con RCT = encendido.

65

YCOCG: conversión RGB a YCOCG se llevó a cabo fuera del códec. Entonces el YCOCG convertido se usó como la entrada para el software JVT.

R+G+B: método propuesto aproximado al comprimir las señales R, G y B por separado.

Y+Co+CG: método propuesto aproximado al comprimir las señales de Y, CO y CG convertidas de forma separada.

JPEG2k_RGB: la compresión JPEG2k se llevó a cabo en el dominio RGB. La transformación de color JPEG2k se apagó.

JPEG2k_YUV: la compresión JPEG2k se llevó a cabo en el dominio YUV. Se usó la transformación de color JPEG2k.

Para el caso solamente interno, el perfil 4:4:4 avanzado propuesto de acuerdo con la presente invención es muy similar al JPEK2k en términos de eficiencia de compresión general. En algunos casos, es inclusive mejor.

El enfoque de acuerdo con los principios de la presente invención es claramente mejor que el perfil 4:4:4 alto actual. A una PSNR igual o mayor que 45dB (PSNR), la mejora promedio en la PSNR promedio es mayor que 1,5 dB. En algunos casos, la mejora se puede traducir en más de 25% de ahorros de bits a una PSNR igual a 45 dB.

Incluso con el mismo tipo de bloque, usar tres modos de predicción es un poco mejor que uno solo. Sin embargo, se pueden utilizar más cambios de sintaxis y semántica.

Ahora se dará una descripción de muchas de las ventajas/característica provistas por los principios de las realizaciones de la presente invención.

Los resultados de prueba demuestran que el perfil 4:4:4 avanzado propuesto, utilizando las mejoras correspondientes a los principios de la presente invención, proporciona un rendimiento mejorado cuando se compara con el perfil 4:4:4 alto actual.

La ganancia de rendimiento es muy importante. Además, mover la transformación de color fuera del códec hará la arquitectura del códec consistente entre todos los otros formatos de color. Como resultado facilitará la implementación y reducirá el costo. También, hará el códec más robusto en términos de seleccionar la transformación óptima de color para alcanzar una eficiencia de codificación mejor. También el enfoque propuesto no añade nuevas herramientas de codificación y requiere solamente unos pequeños cambios en la sintaxis y en la semántica. Se proporcionan modificaciones para la norma H.264 existente que mejoran el rendimiento más allá de lo actualmente alcanzable. Además, el rendimiento se mejora más allá de JPEG-2000 para aplicaciones de alta calidad. De acuerdo con un ejemplo de antecedente ilustrativo no reivindicado, se pueden lograr muchas mejoras en el rendimiento de codificación 4:4:4 en la norma H.264 usando el algoritmo de codificación de luma para codificar los tres componentes de color del contenido 4:4:4. Esto es, no son necesarias nuevas herramientas para el algoritmo de compresión de luma (o croma, que no se usa). En su lugar, se utilizan las herramientas de codificación de luma. Además, los cambios en semántica y sintaxis en el perfil 4:4:4 actual se pueden implementar de acuerdo con los presentes principios para dar soporte a la codificación de luma de los tres canales de componente. En las pruebas conducidas de acuerdo con una realización de la presente invención, cuando el contenido de origen tiene muchas texturas y bordes espaciales, las herramientas de predicción espacial usadas en luma, claramente exhiben un rendimiento superior a las usadas en croma. Para algunas secuencias de prueba, cuando cada componente de color se codificó como luma, más de un 30% en la reducción de bits se observó a una calidad comprimida mayor que o igual que 45 dB (PSNR promedio).

Estas y otras características y ventajas de la presente invención pueden alcanzarse fácilmente por expertos en la técnica basándose en las enseñanzas de la misma. Se debe entender que las enseñanzas de la presente invención se pueden implementar en diferentes formas de hardware, software, firmware, procesadores de propósitos especiales o combinaciones de los mismos.

Más preferiblemente, las enseñanzas de la presente invención se pueden implementar como una combinación de hardware y software. Además, el software puede implementarse como un programa de aplicación incorporado tangiblemente en una unidad de almacenamiento de programa. El programa de aplicación se puede cargar, ejecutar por una máquina que comprenda la arquitectura apropiada. Preferiblemente, la máquina se puede implementar en una plataforma informática con un hardware como una o más unidades de procesamiento central ("CPU"), una memoria de acceso aleatorio ("RAM"), e interfaces de salida y entrada ("I/O"). La plataforma informática también puede incluir un sistema operativo y un código de microinstrucciones. Los diferentes procesos y funciones descritos en el presente documento pueden ser parte del código de microinstrucción o parte del programa de aplicación o una combinación de los mismos, los cuales se pueden ejecutar con una CPU. Además, se pueden conectar otras unidades periféricas con la plataforma informática como una unidad de almacenamiento de datos adicional y una unidad de impresión.

También, se debe entender que debido a que algunos componentes y métodos del sistema representados en los dibujos adjuntos preferiblemente se implementan en un software, las conexiones reales entre los componentes del sistema o los bloques de función del proceso pueden diferir dependiendo de la manera en que se programe la presente invención. Habiendo proporcionado las enseñanzas de la misma, los expertos en la técnica podrán contemplar estas y otras implementaciones o configuraciones similares de la presente invención.

Aunque se han descrito las realizaciones ilustrativas en el presente documento con referencia a los dibujos adjuntos, se debe entender que la presente invención no está limitada a esas precisas realizaciones, y que se pueden efectuar varios cambios y modificaciones por los expertos en la técnica sin apartarse del alcance de la presente invención como se establece en las reivindicaciones adjuntas.

REIVINDICACIONES

1.- Un aparato para decodificar datos de señales de video para un bloque de imagen, comprendiendo el aparato:

- 5 un decodificador (200) para decodificar componentes de color de los datos de señal de video, en el que dicho decodificador (200) está configurado para decodificar todos los componentes de color de los datos de señal de video usando predictores que se determinan independientemente para cada uno de los componentes de color y en el que
- 10 el muestreo de los datos de señal de video corresponde a un formato 4:4:4, en el que dicho decodificador está configurado para decodificar todos los componentes de color de los datos de señal de video usando predictores espaciales determinados independientemente para la intratrama y usando los mismos filtros de interpolación para las tramas B y P para calcular píxeles de referencia en ubicaciones de píxeles fraccionales, el decodificador estando configurado para decodificar los componentes de color de los datos de señal de vídeo sin aplicar una transformación de color residual a los mismos, y el aparato comprendiendo además un módulo (293) de transformación de color
- 15 inversa, de posdescodificador, en comunicación de señal con dicho decodificador, para realizar selectivamente una transformación de color inversa en componentes de color de los datos de señal de vídeo después de la decodificación de los componentes de color por dicho decodificador.

20 2.- El aparato de la reivindicación 1, en el que la transformación de color inversa realizada por dicho módulo (293) de transformación de color inversa de posdescodificador es una transformación de color no residual.

3.- El aparato de la reivindicación 1, en el que dicho módulo (293) de transformación de color inversa de posdescodificador está configurado para realizar la transformación de color inversa en las imágenes de origen correspondientes a los datos de señal de video.

25 4.- El aparato de la reivindicación 1, en el que dicho módulo (293) de transformación de color inversa de posdescodificador está configurado para proporcionar componentes de color transformados de los datos de señal de video en el formato RGB o cualquier formato no RGB.

30 5.- Un método para decodificar datos de señal de video para un bloque de imagen, el método comprendiendo:

decodificar (1500) componentes de color de los datos de señal de video, en el que dicho paso de decodificación decodifica todos los componentes de color transformados de los datos de señal de video usando predictores que se determinan independientemente para cada uno de los componentes (1530) de color y en el que el muestreo de los

35 datos de señal de video corresponde a un formato 4:4:4, en el que todos los componentes de color de los datos de señal de video se decodifican usando predictores espaciales determinados independientemente para la intratrama y usando los mismos filtros de interpolación para las tramas B y P para calcular píxeles de referencia en ubicaciones de píxeles fraccionales, en el que la decodificación de los componentes de color de los datos de señal de vídeo se realiza sin aplicar una transformación de color residual a los mismos, y en el que el método comprende además un

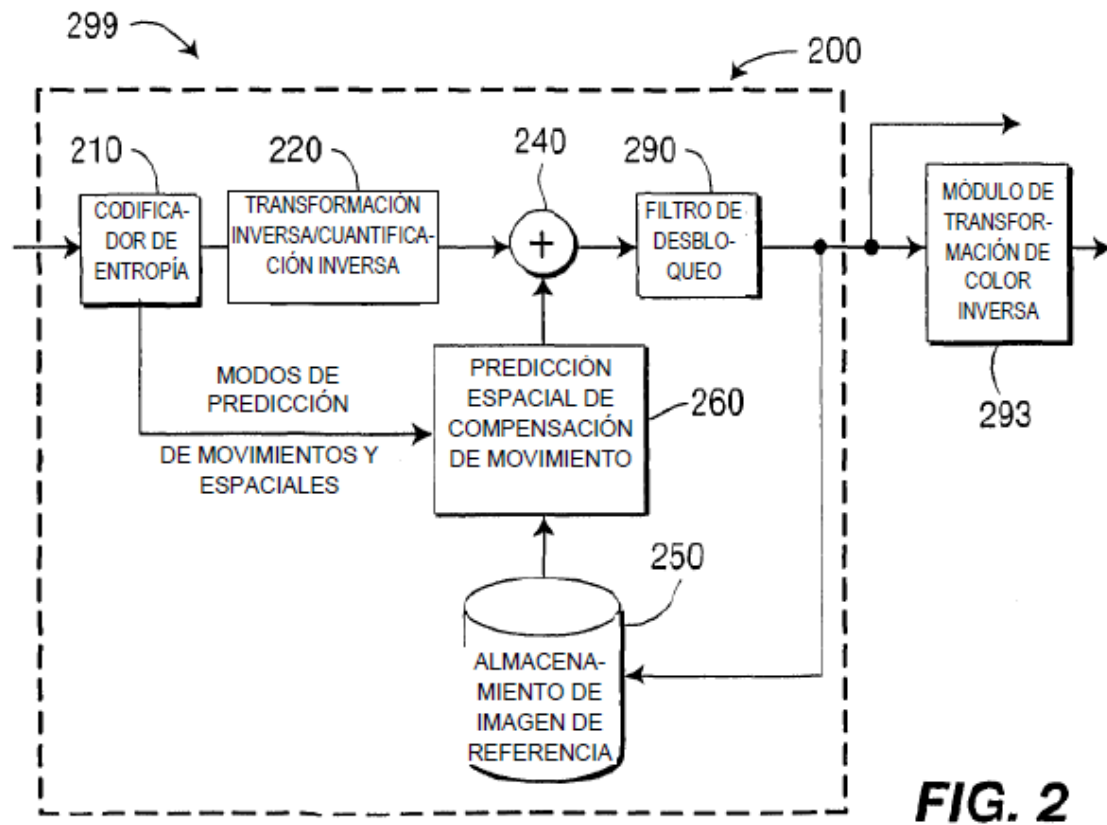
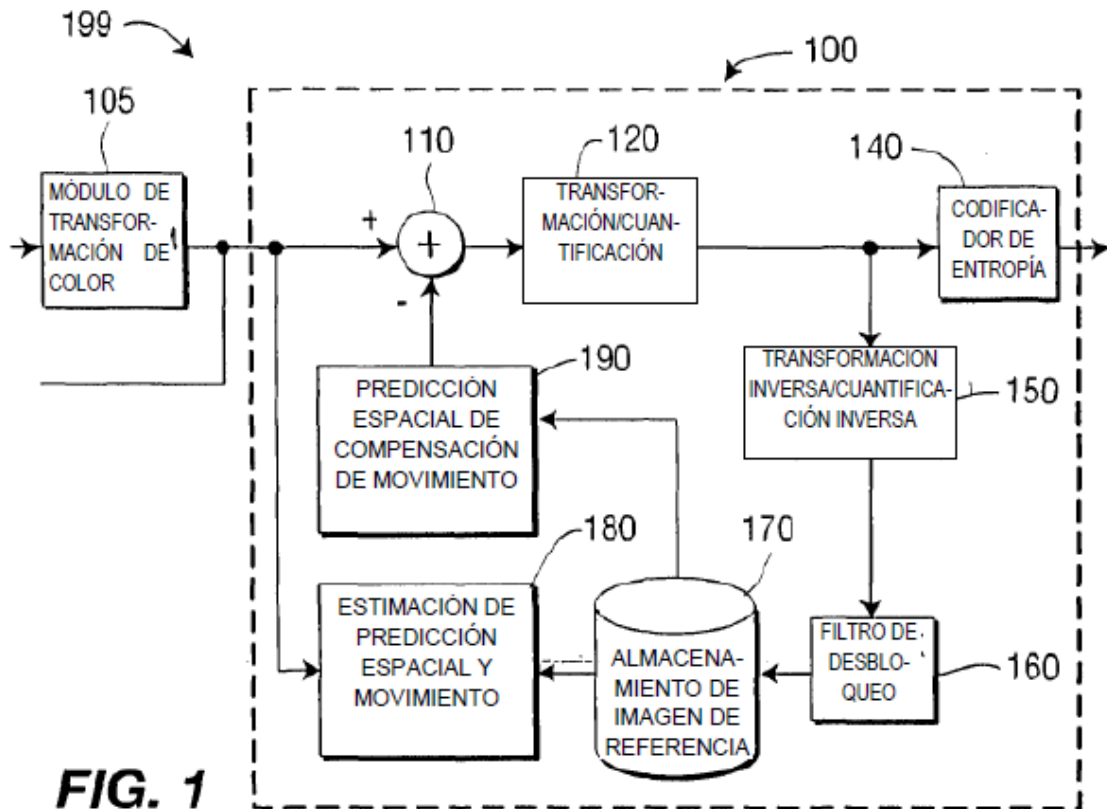
40 paso (1564) para realizar selectivamente una transformación de color inversa en componentes de color decodificados de los datos de señal de vídeo.

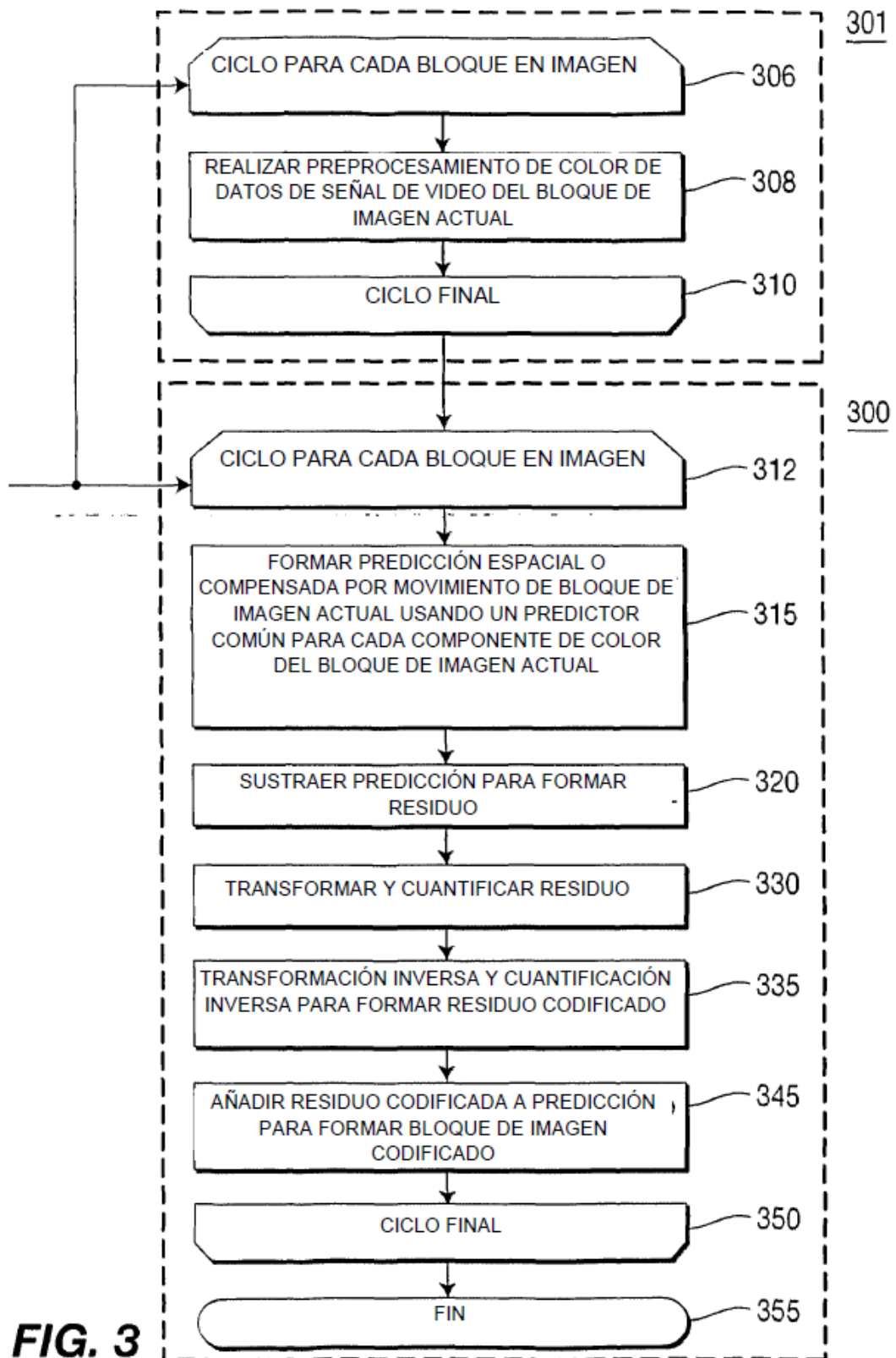
6.- El método de la reivindicación 5, en el que la transformación de color inversa realizada por dicho paso (1564) es una transformación de color no residual.

45 7.- El método de la reivindicación 5, en el que dicho paso (1564) realiza la transformación de color inversa en las imágenes de origen correspondientes a los datos de señal de video.

8.- El método de la reivindicación 5, en el que dicho paso (1564) proporciona componentes de color transformados de los datos de señal de vídeo en el formato RGB o cualquier formato no RGB.

50





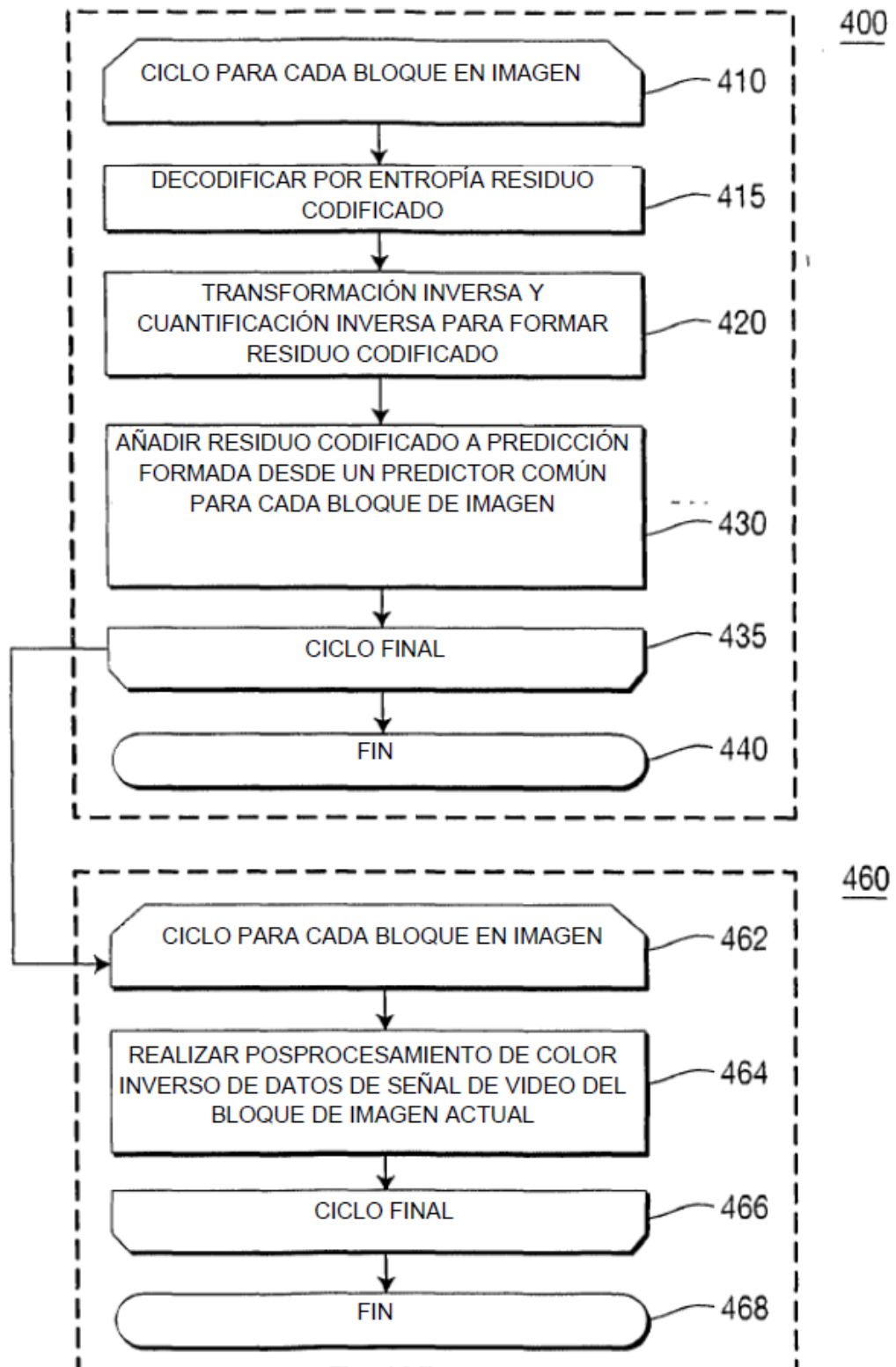
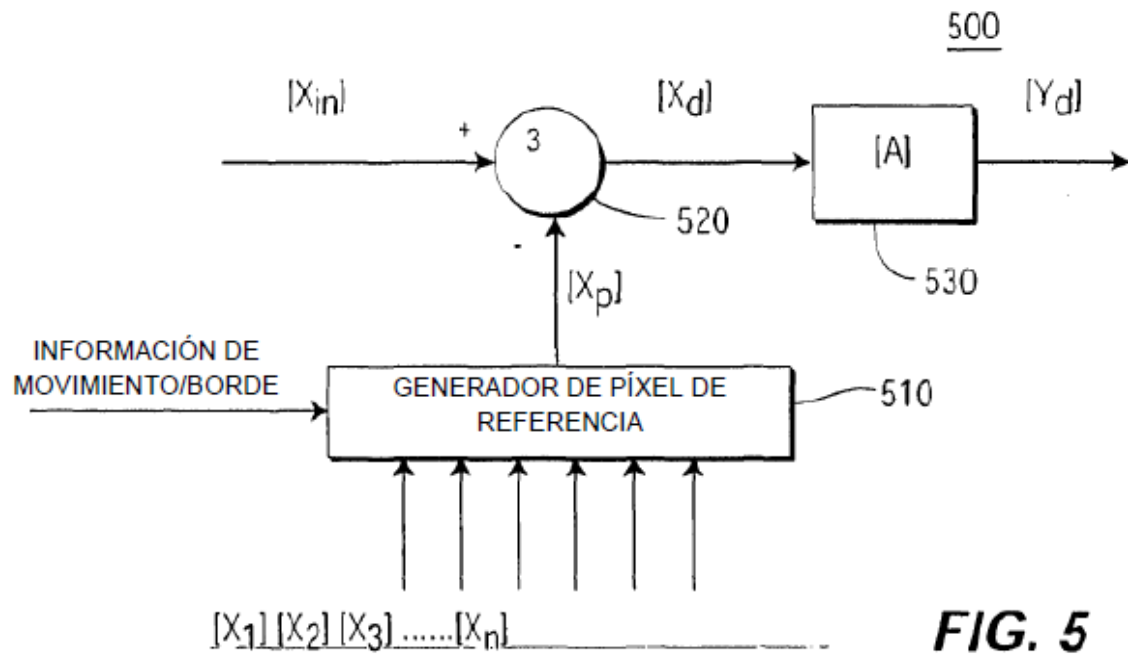


FIG. 4



1200

seq_parameter_set_rbsp() {	C	Descriptor
profile_idc	0	u(8)
constraint_set0_flag	0	u(1)
constraint_set1_flag	0	u(1)
constraint_set2_flag	0	u(1)
constraint_set3_flag	0	u(1)
reserved_zero_4bits /* equal to 0 */	0	u(4)
level_idc	0	u(8)
seq_parameter_set_id	0	ue(v)
if(profile_idc == 100 profile_idc == 110 profile_idc == 122 profile_idc == 166 profile_idc == 188) {		
chroma_format_idc	0	ue(v)
bit_depth_luma_minus8	0	ue(v)

FIG. 12

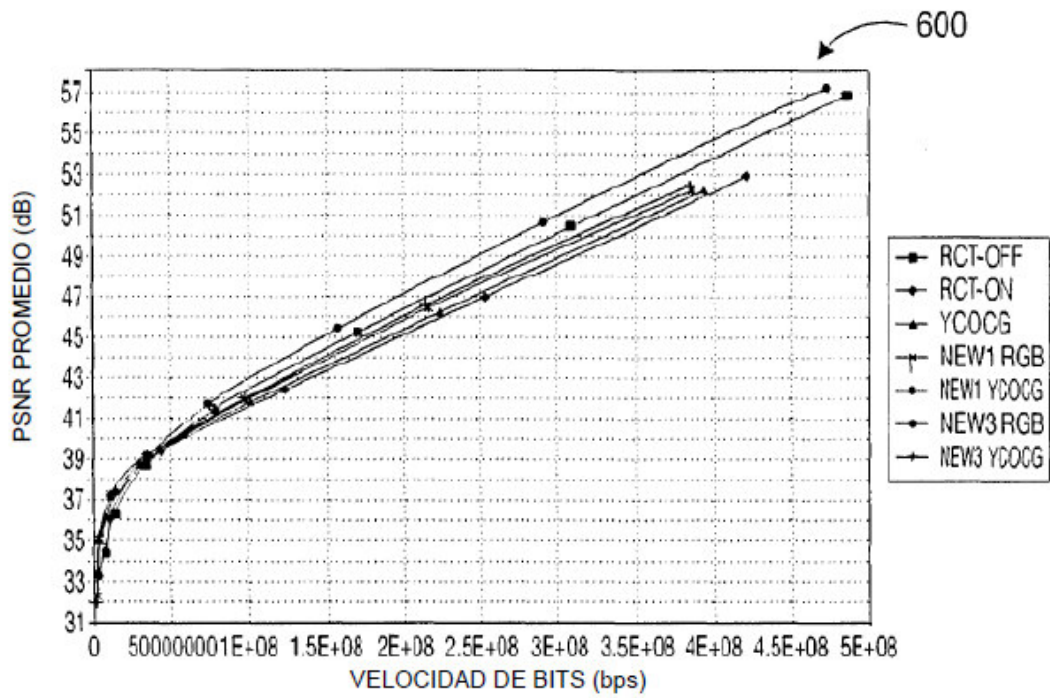


FIG. 6A

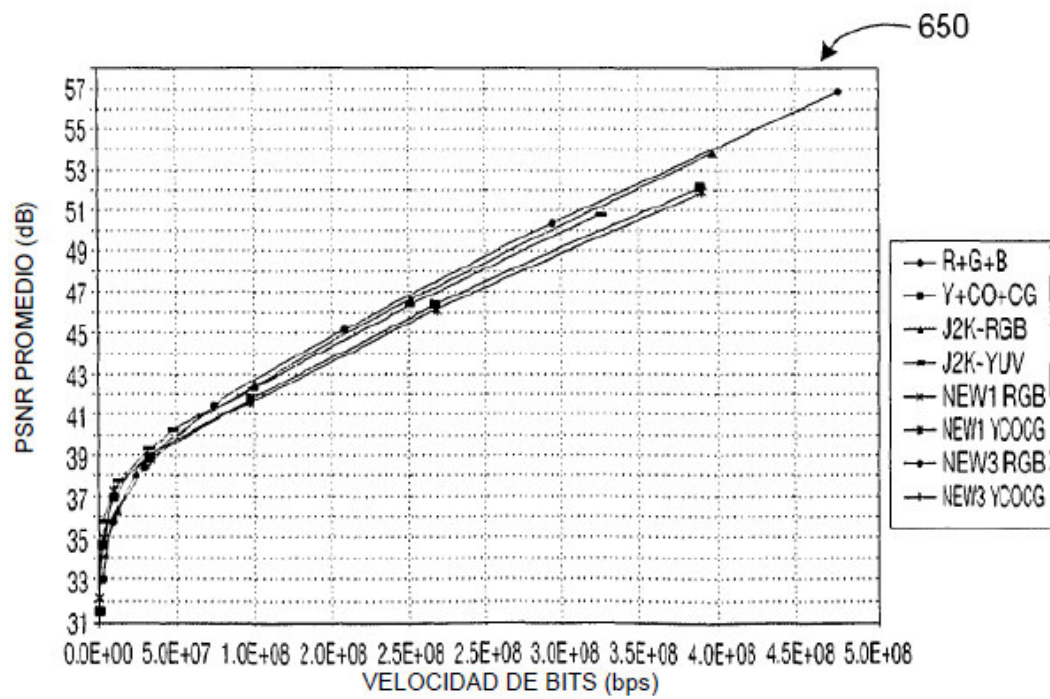


FIG. 6B

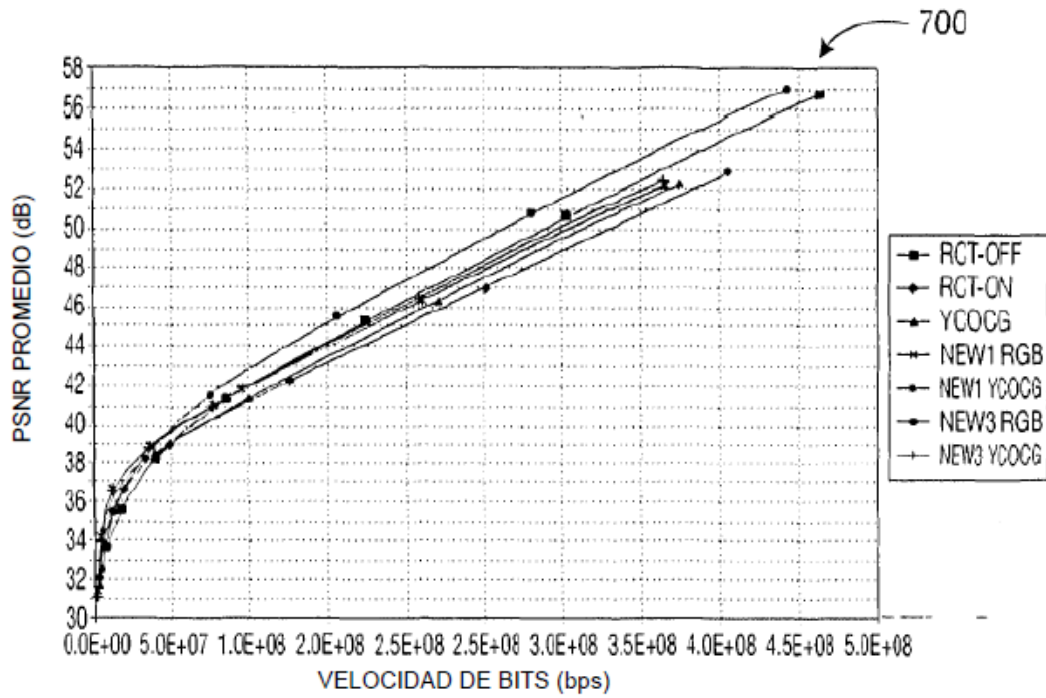


FIG. 7A

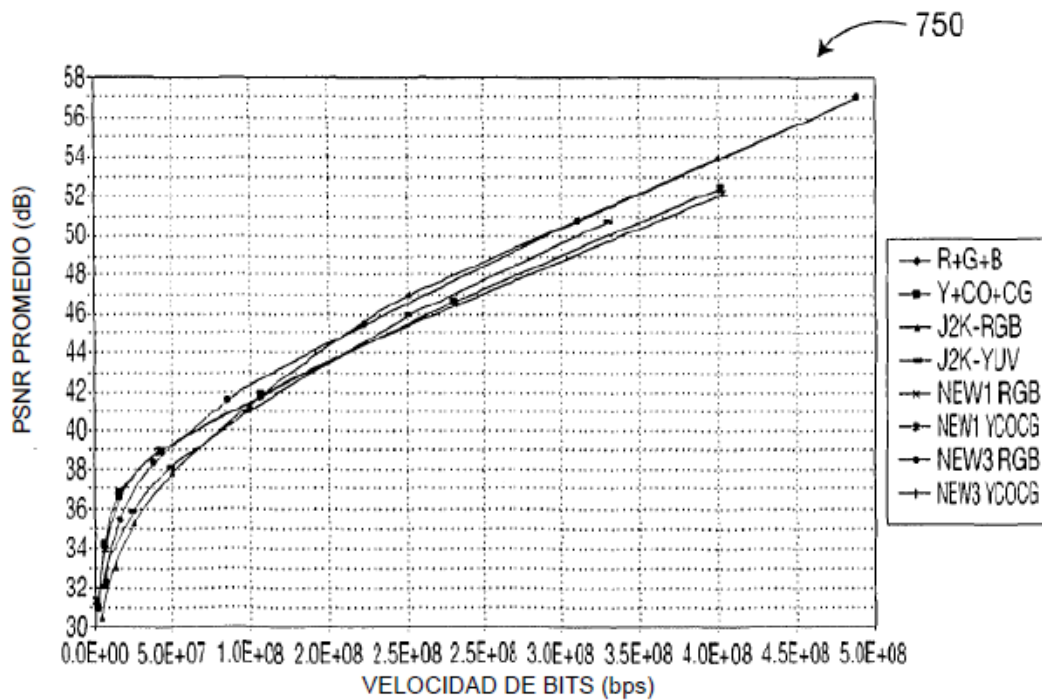


FIG. 7B

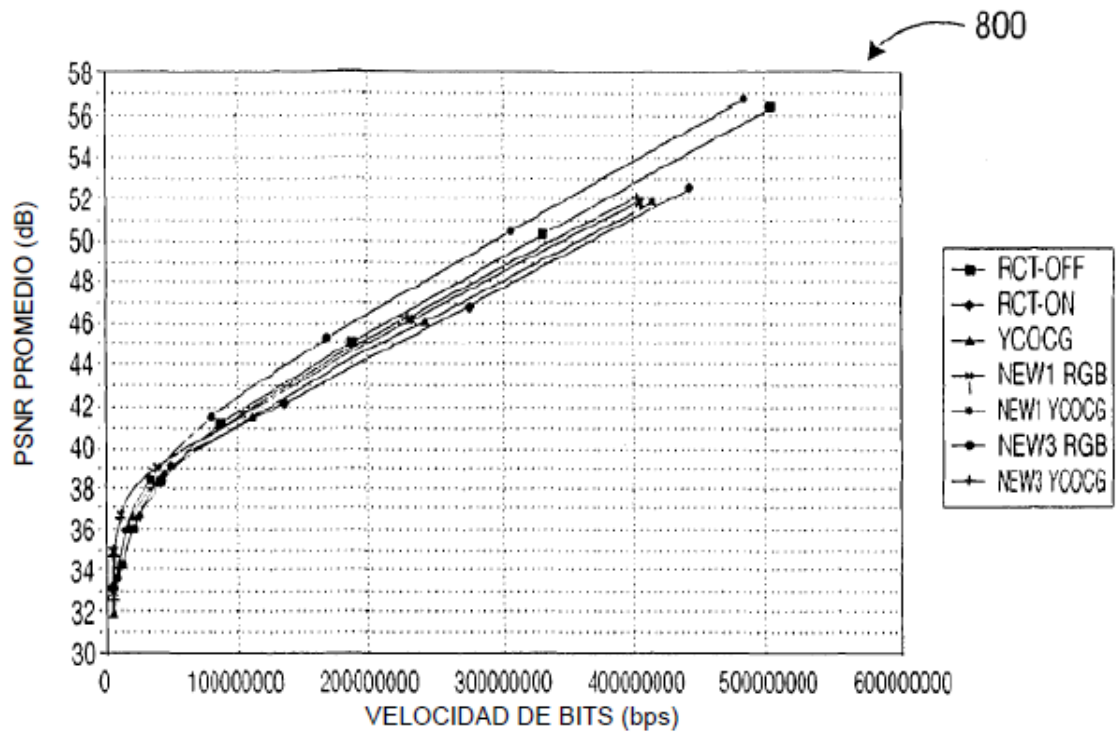


FIG. 8A

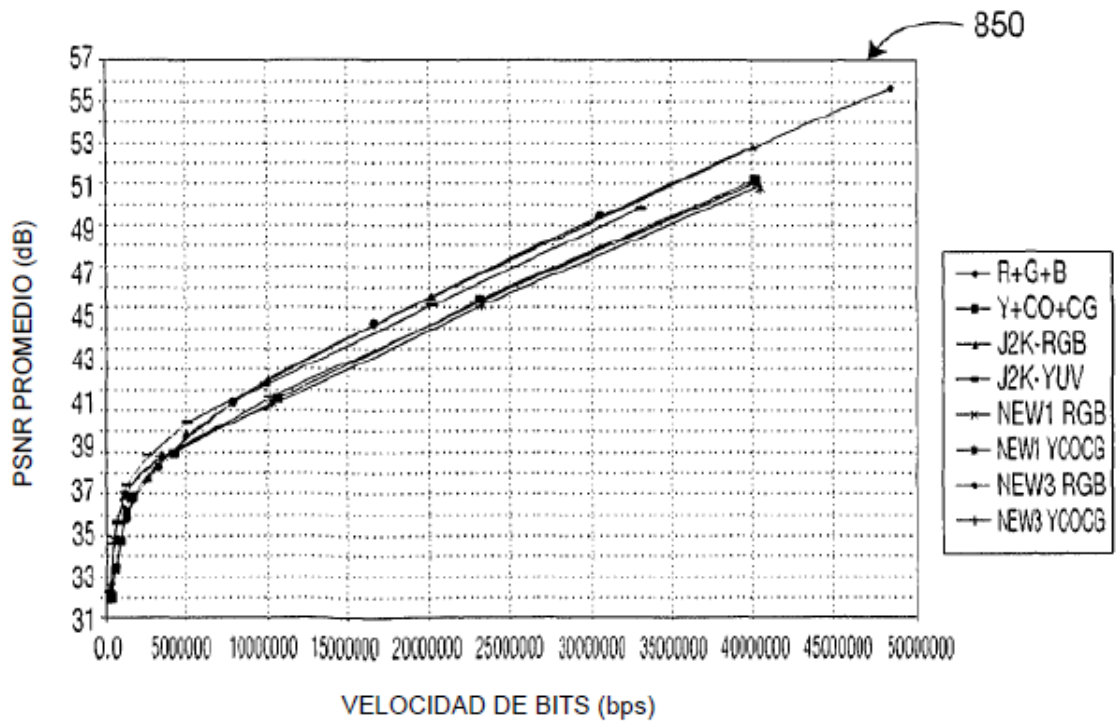


FIG. 8B

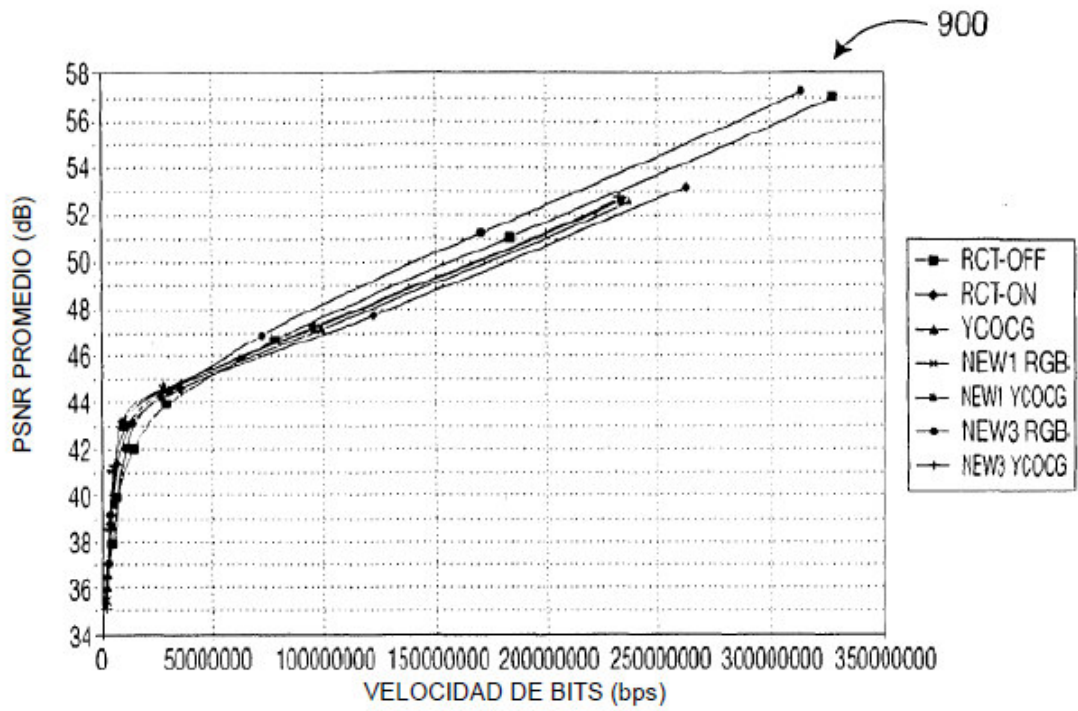


FIG. 9A

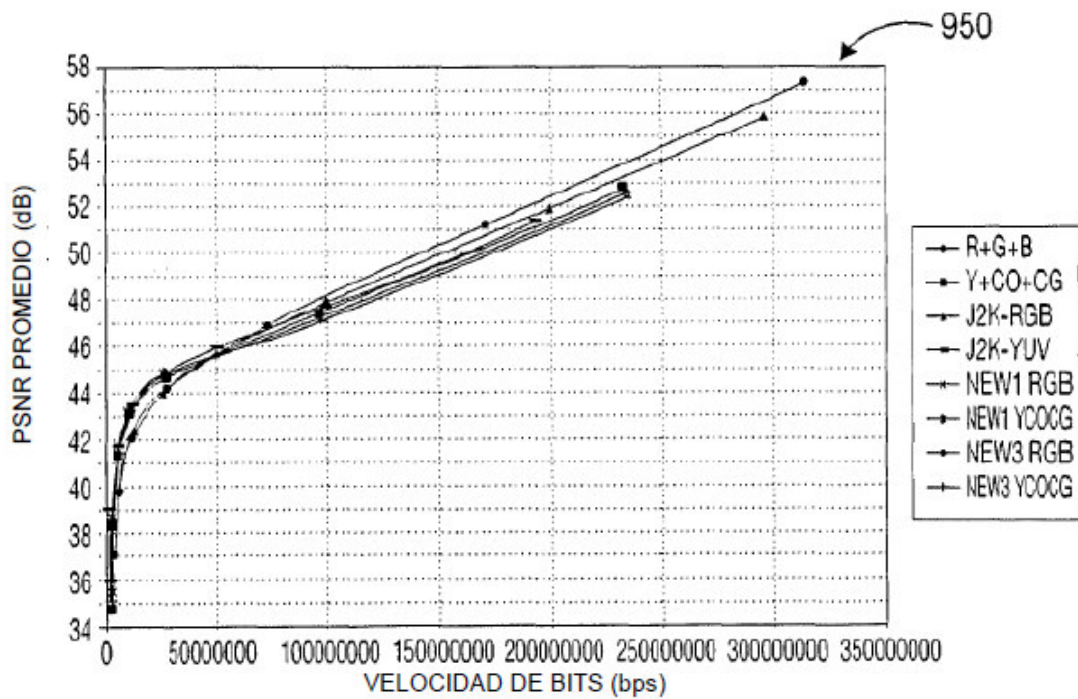


FIG. 9B

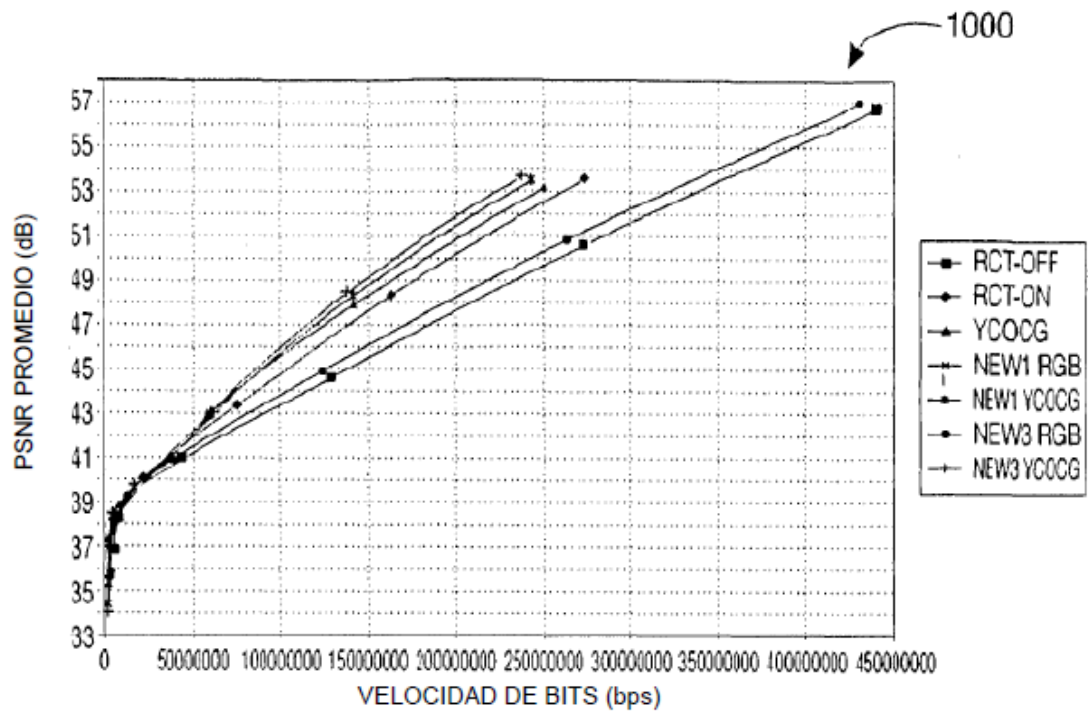


FIG. 10A

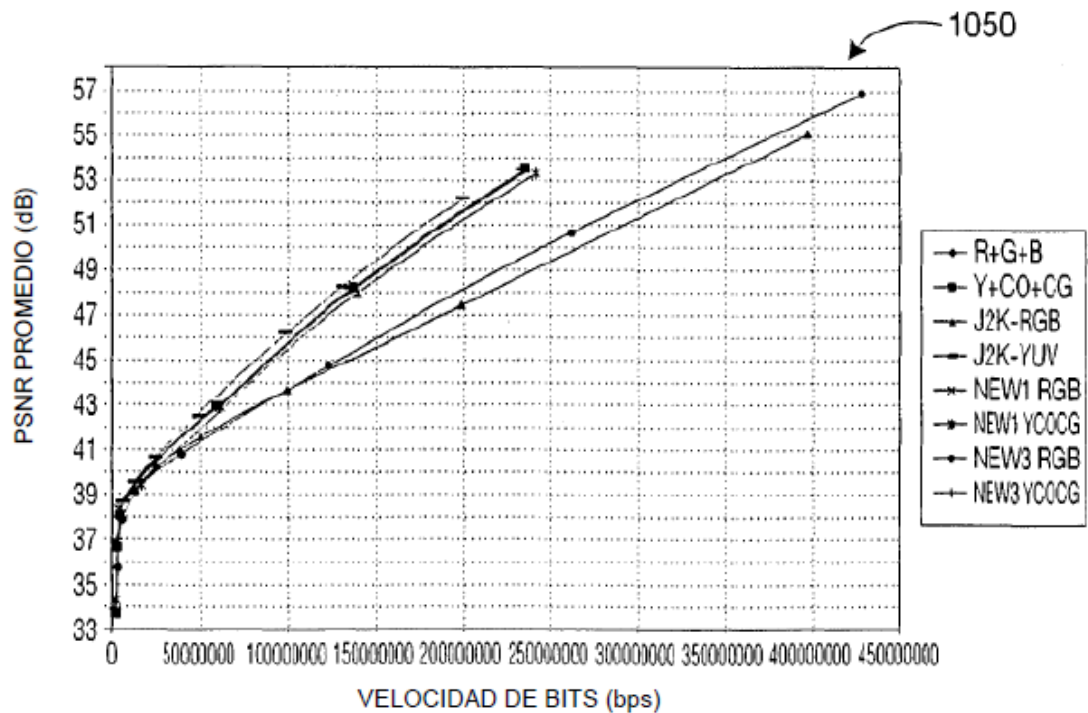


FIG. 10B

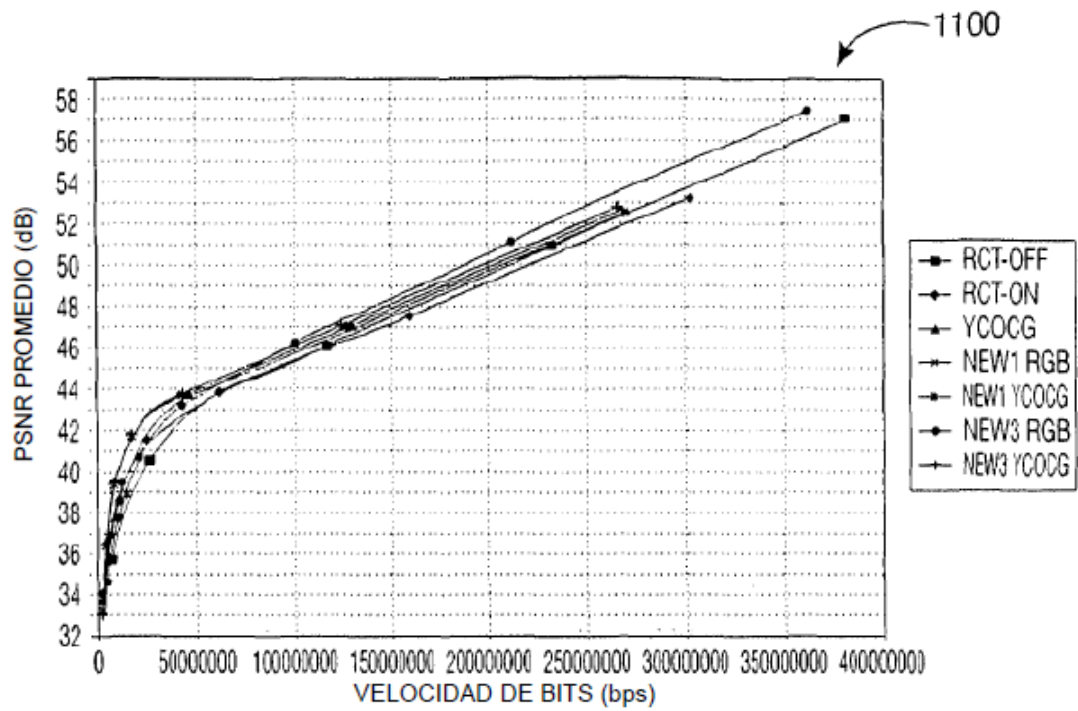


FIG. 11A

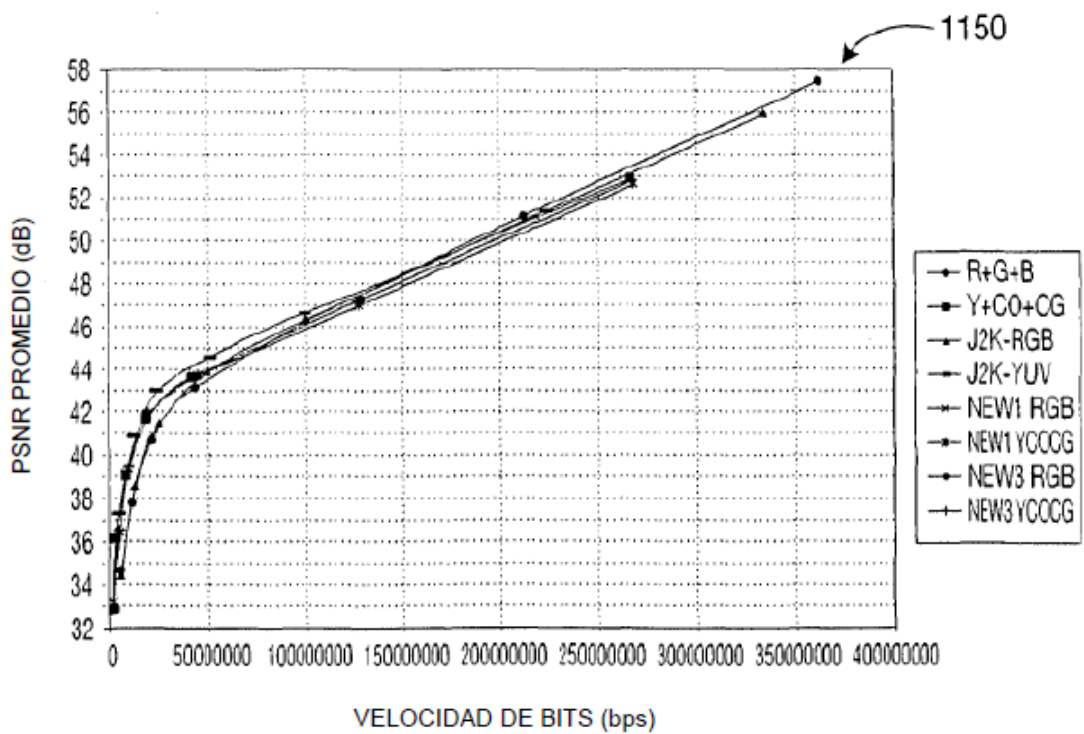


FIG. 11B

FIG. 13**FIG. 13A****FIG. 13B****FIG. 13C****FIG. 13D****FIG. 13A**

Residual() {	C	Descriptor
if(!entropy_coding_mode_flag)		
residual_block = residual_block_cavlc		
else		
residual_block = residual_block_cabac		
if(MbPartPredMode(mb_type, 0) == Intra_16x16)		
residual_block(Intra16x16DCLLevel, 16)	3	
for(i8x8 = 0; i8x8 < 4; i8x8++) /* each luma 8x8 block */		
if(!transform_size_8x8_flag !entropy_coding_mode_flag)		
for(i4x4 = 0; i4x4 < 4; i4x4++) { /* each 4x4 sub-block of block */		
if(CodedBlockPatternLuma & (1 << i8x8))		
if(MbPartPredMode(mb_type, 0) == Intra_16x16)		
residual_block(Intra16x16ACLevel[i8x8 * 4 + i4x4], 15)	3	
else		
residual_block(LumaLevel[i8x8 * 4 + i4x4], 16)	3 4	
else if(MbPartPredMode(mb_type, 0) == Intra_16x16)		
for(i = 0; i < 15; i++)		
Intra16x16ACLevel[i8x8 * 4 + i4x4][i] = 0		
else		
for(i = 0; i < 16; i++)		
LumaLevel[i8x8 * 4 + i4x4][i] = 0		
if(!entropy_coding_mode_flag && transform_size_8x8_flag)		
for(i = 0; i < 16; i++)		
LumaLevel8x8[i8x8][4 * i + i4x4] =		
LumaLevel[i8x8 * 4 + i4x4][i]		
}		

FIG. 13B

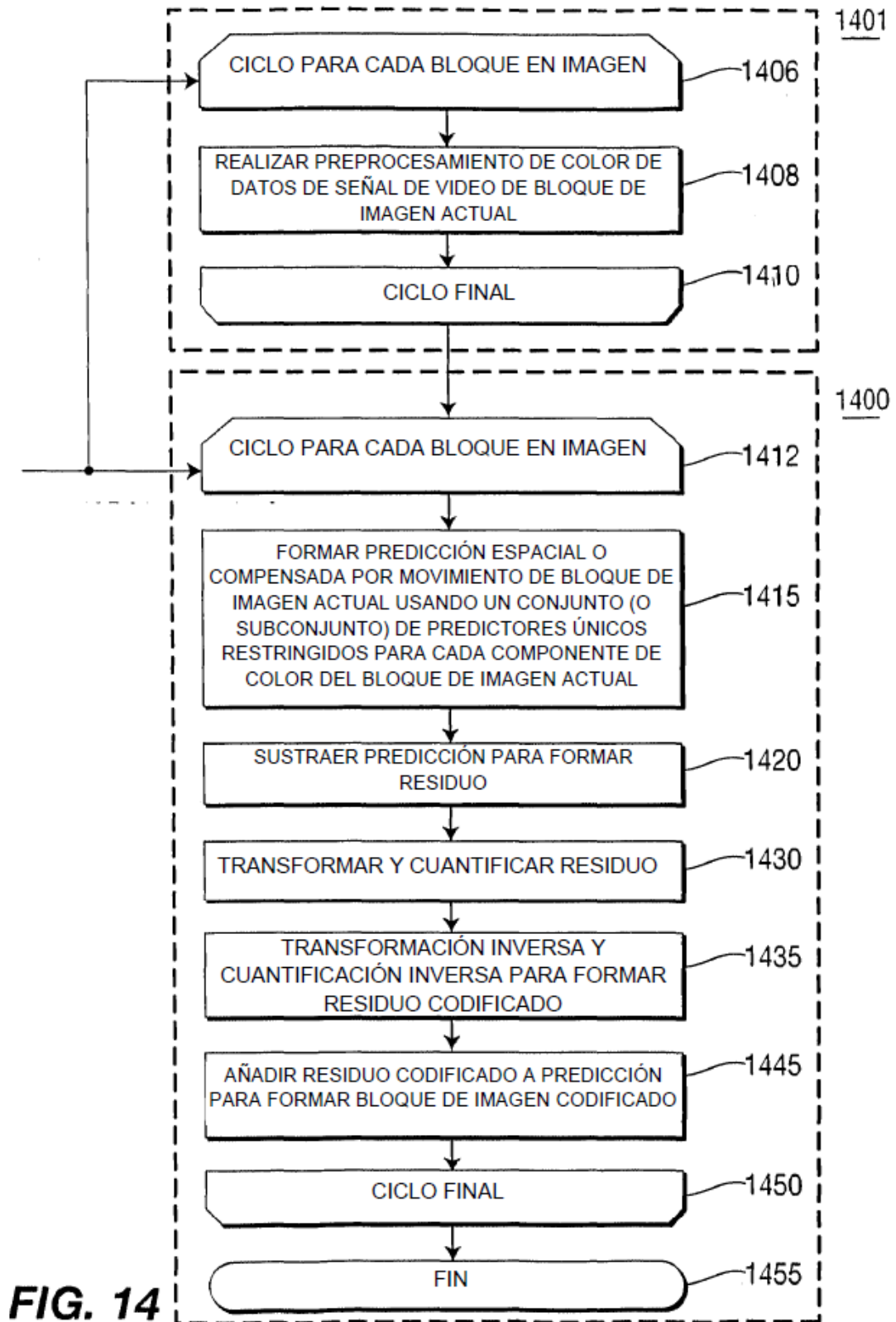
else if(CodedBlockPatternLuma & (1 << i8x8))		
residual_block(LumaLevel8x8[i8x8], 64)	3 4	
else		
for(i = 0; i < 64; i++)		
LumaLevel8x8[i8x8][i] = 0		
if(chroma_format_idc == 1 chroma_format_idc == 2) {		
NumC8x8 = 4 / (SubWidthC * SubHeightC)		
for(iCbCr = 0; iCbCr < 2; iCbCr++)		
if(CodedBlockPatternChroma & 3) /* chroma DC residual present */		
residual_block(ChromaDCLevel[iCbCr], 4 * NumC8x8)	3 4	
else		
for(i = 0; i < 4 * NumC8x8; i++)		
ChromaDCLevel[iCbCr][i] = 0		
for(iCbCr = 0; iCbCr < 2; iCbCr++)		
for(i8x8 = 0; i8x8 < NumC8x8; i8x8++)		
for(i4x4 = 0; i4x4 < 4; i4x4++)		
if(CodedBlockPatternChroma & 2)		
/* chroma AC residual present */		
residual_block(ChromaACLevel[iCbCr][i8x8*4+i4x4], 15)	3 4	
else		
for(i = 0; i < 15; i++)		
ChromaACLevel[iCbCr][i8x8*4+i4x4][i] = 0		
} else if(chroma_format_idc == 3) {		
if(MbPartPredMode(mb_type, 0) == Intra_16x16) /* Cb for 4:4:4 */		
residual_block(CbIntra16x16DCLevel, 16)	3	
for(i8x8 = 0; i8x8 < 4; i8x8++) /* each Cb 8x8 block */		
if(!transform_size_8x8_flag !entropy_coding_mode_flag)		
for(i4x4 = 0; i4x4 < 4; i4x4++) { /* each 4x4 sub-block of block		
*/		
if(CodedBlockPatternLuma & (1 << i8x8))		

FIG. 13C

if(MbPartPredMode(mb_type, 0) == Intra_16x16)		
residual_block(CbIntra16x16ACLevel[i8x8*4 + i4x4], 15)	3	
else		
residual_block(CbLevel[i8x8 * 4 + i4x4], 16)	3 4	
else if(MbPartPredMode(mb_type, 0) == Intra_16x16)		
for(i = 0; i < 15; i++)		
CbIntra16x16ACLevel[i8x8 * 4 + i4x4][i] = 0		
else		
for(i = 0; i < 16; i++)		
CbLevel[i8x8 * 4 + i4x4][i] = 0		
if(!entropy_coding_mode_flag && transform_size_8x8_flag)		
for(i = 0; i < 16; i++)		
CbLevel8x8[i8x8][4 * i + i4x4] = CbLevel[i8x8 * 4 + i4x4][i]		
}		
else if(CodedBlockPatternLuma & (1 << i8x8))		
residual_block(CbLevel8x8[i8x8], 64)	3 4	
else		
for(i = 0; i < 64; i++)		
CbLevel8x8[i8x8][i] = 0		
if(MbPartPredMode(mb_type, 0) == Intra_16x16) /*Cr for 4:4:4 */		
residual_block(CrIntra16x16DCLevel, 16)	3	
for(i8x8 = 0; i8x8 < 4; i8x8++) /* each Cr 8x8 block */		
if(!transform_size_8x8_flag !entropy_coding_mode_flag)		
for(i4x4 = 0; i4x4 < 4; i4x4++) { /* each 4x4 sub-block of block */		
if(CodedBlockPatternLuma & (1 << i8x8))		
if(MbPartPredMode(mb_type, 0) == Intra_16x16)		
residual_block(CrIntra16x16ACLevel[i8x8*4+i4x4], 15)	3	
else		

FIG. 13D

residual_block(CrLevel[i8x8 * 4 + i4x4], 16)	3 4	
else if(MbPartPredMode(mb_type, 0) == Intra_16x16)		
for(i = 0; i < 15; i++)		
CrIntra16x16ACLevel[i8x8 * 4 + i4x4][i] = 0		
else		
for(i = 0; i < 16; i++)		
CrLevel[i8x8 * 4 + i4x4][i] = 0		
if(!entropy_coding_mode_flag && transform_size_8x8_flag)		
for(i = 0; i < 16; i++)		
CrLevel8x8[i8x8][4 * i + i4x4] = CrLevel[i8x8 * 4 + i4x4][i]		
}		
else if(CodedBlockPatternLuma & (1 << i8x8))		
residual_block(CrLevel8x8[i8x8], 64)	3 4	
else		
for(i = 0; i < 64; i++)		
CrLevel8x8[i8x8][i] = 0		
}		



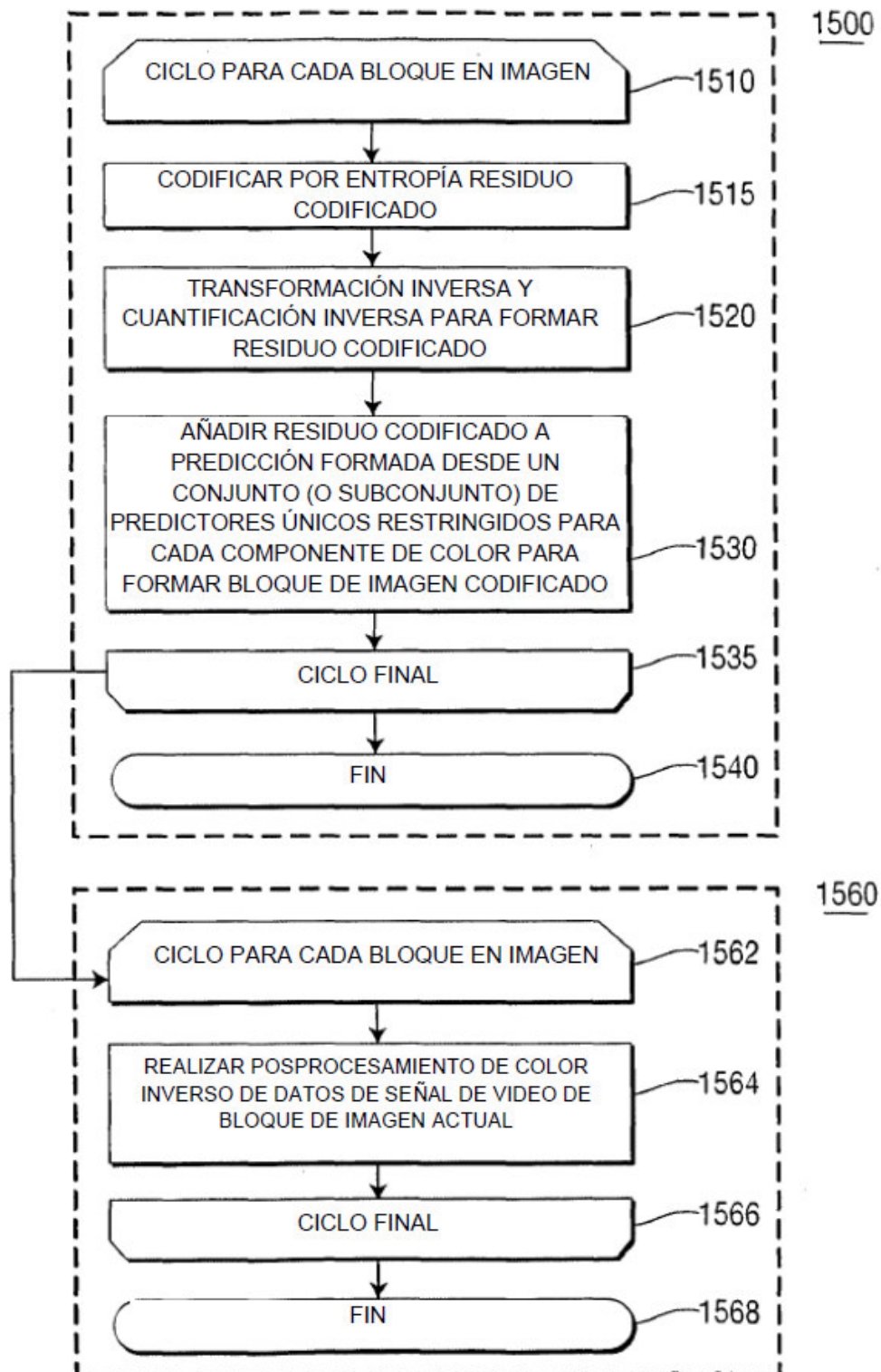


FIG. 15

FIG. 16**FIG. 16A****FIG. 16B**

1600

FIG. 16A

mb_pred(mb_type) {	C	Descriptor
if(MbPartPredMode(mb_type, 0) == Intra_4x4 MbPartPredMode(mb_type, 0) == Intra_8x8 MbPartPredMode(mb_type, 0) == Intra_16x16) {		
if(profile_idc != 166) {		
if(MbPartPredMode(mb_type, 0) == Intra_4x4)		
for(luma4x4BlkIdx=0; luma4x4BlkIdx<16; luma4x4BlkIdx++) {		
prev_intra4x4_pred_mode_flag[luma4x4BlkIdx]	2	u(1) ae(v)
if(!prev_intra4x4_pred_mode_flag[luma4x4BlkIdx])		
rem_intra4x4_pred_mode[luma4x4BlkIdx]	2	u(3) ae(v)
}		
if(MbPartPredMode(mb_type, 0) == Intra_8x8)		
for(luma8x8BlkIdx=0; luma8x8BlkIdx<4; luma8x8BlkIdx++) {		
prev_intra8x8_pred_mode_flag[luma8x8BlkIdx]	2	u(1) ae(v)
if(!prev_intra8x8_pred_mode_flag[luma8x8BlkIdx])		
rem_intra8x8_pred_mode[luma8x8BlkIdx]	2	u(3) ae(v)
}		
if(chroma_format_idc != 0)		
intra_chroma_pred_mode	2	u(v) ae(v)
} else {		
if(MbPartPredMode(mb_type, 0) == Intra_4x4)		
for(luma4x4BlkIdx=0; luma4x4BlkIdx<16; luma4x4BlkIdx++) {		
prev_intra4x4_pred_mode_flag0[luma4x4BlkIdx]	2	u(1) ae(v)
if(!prev_intra4x4_pred_mode_flag0[luma4x4BlkIdx])		
rem_intra4x4_pred_mode0[luma4x4BlkIdx]	2	u(3) ae(v)
prev_intra4x4_pred_mode_flag1[luma4x4BlkIdx]	2	u(1) ae(v)

FIG. 16B

if(!prev_intra4x4_pred_mode_flag[luma4x4BlkIdx])		
rem_intra4x4_pred_mode1[luma4x4BlkIdx]	2	u(3) ae(v)
prev_intra4x4_pred_mode_flag2[luma4x4BlkIdx]	2	u(1) ae(v)
if(!prev_intra4x4_pred_mode_flag[luma4x4BlkIdx])		
rem_intra4x4_pred_mode2[luma4x4BlkIdx]	2	u(3) ae(v)
}		
if(MbPartPredMode(mb_type, 0) == Intra_8x8)		
for(luma8x8BlkIdx=0; luma8x8BlkIdx<4; luma8x8BlkIdx++) {		
prev_intra8x8_pred_mode_flag0[luma8x8BlkIdx]	2	u(1) ae(v)
if(!prev_intra8x8_pred_mode_flag0[luma8x8BlkIdx])		
rem_intra8x8_pred_mode0[luma8x8BlkIdx]	2	u(3) ae(v)
prev_intra8x8_pred_mode_flag1[luma8x8BlkIdx]	2	u(1) ae(v)
if(!prev_intra8x8_pred_mode_flag1[luma8x8BlkIdx])		
rem_intra8x8_pred_mode1[luma8x8BlkIdx]	2	u(3) ae(v)
prev_intra8x8_pred_mode_flag2[luma8x8BlkIdx]	2	u(1) ae(v)
if(!prev_intra8x8_pred_mode_flag2[luma8x8BlkIdx])		
rem_intra8x8_pred_mode2[luma8x8BlkIdx]	2	u(3) ae(v)
}		
}		
} else if(MbPartPredMode(mb_type, 0) != Direct) {		
.....		