

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 805 105**

51 Int. Cl.:

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

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **16.03.2006 PCT/US2006/009381**
- 87 Fecha y número de publicación internacional: **26.10.2006 WO06112992**
- 96 Fecha de presentación y número de la solicitud europea: **16.03.2006 E 06738444 (6)**
- 97 Fecha y número de publicación de la concesión europea: **17.06.2020 EP 1872586**

54 Título: **Codificación luma-croma con tres predictores espaciales distintos**

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:
10.02.2021

73 Titular/es:

DOLBY INTERNATIONAL AB (100.0%)
Apollo Building, 3E, Herikerbergweg 1-35
1101 CN Amsterdam Zuidoost, NL

72 Inventor/es:

YU, HAOPING y
LIU, LIMIN

74 Agente/Representante:

ELZABURU, S.L.P

ES 2 805 105 T3

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

DESCRIPCIÓN

Codificación luma-croma con tres predictores espaciales distintos

Referencia a solicitudes relacionadas

5 Esta solicitud reivindica el beneficio de la solicitud provisional de EE.UU. con número de serie 60/671,255, presentada el 13 de abril de 2005, y de la solicitud provisional de EE.UU. con número de serie 60/700,834, presentada el 20 de julio de 2005. Además, esta solicitud está relacionada con las solicitudes de patente de EE.UU. de expedientes números PU050051, PU060023 tituladas "METHOD AND APPARATUS FOR VIDEO ENCODING"; y PU060029, PU060030 tituladas "METHOD AND APPARATUS FOR VIDEO DECODING", cada una presentada simultáneamente con la presente.

10 **Sector técnico de la invención**

La presente invención se refiere, en general, a codificadores y decodificadores de video y, más particularmente, a procedimientos y aparatos para codificación y decodificación de video.

Antecedentes de la invención

15 Actualmente, el formato 4:4:4 del estándar H.264 de la Unión Internacional de Telecomunicaciones, sector de telecomunicaciones (ITU-T, International Telecommunication Union, Telecommunication Sector) (en adelante, el "estándar H.264") codifica solamente uno de los tres canales como luma, siendo codificados los otros dos canales como croma utilizando herramientas menos eficientes. Cuando una entrada a un códec está en el formato 4:4:4 con plena resolución en cada componente de entrada, codificar dos de tres componente de entrada con el algoritmo de
20 codificación croma menos efectivo tiene como resultado la utilización de más bits en esos dos canales. Este problema particular es más notorio en cuadros intra. Por ejemplo, el estándar H.264 funcionando en modo solamente intra es menos eficiente que JPEG2k para calidad de compresión global a 40 dB (PSNR) y por encima.

Por consiguiente, sería deseable y muy ventajoso tener procedimientos y aparatos para codificación y decodificación de video que superen los inconvenientes de la técnica anterior mencionados anteriormente.

25 El documento "Proposal for the unsolved issues in Professional Extensions II", una contribución a JVT de ISO/IEC MPEG & ITU-T VCEG para la décima reunión JVT, 2003, da a conocer una predicción intra-croma con un predictor común para casos luma y croma y que permite, más en general, que los casos croma utilicen los mismos predictores que se conocen para los casos luma. Da a conocer asimismo un caso en el que para un video YUV con una alta resolución, se utiliza la misma transformada bilineal como un filtro de interpolación de compensación de movimiento común.

30 **Características de la invención**

Estos y otros inconvenientes y desventajas de la técnica anterior son tratados por la presente invención, que está dirigida a procedimientos y aparatos para codificación y decodificación de video, según las reivindicaciones 1, 3, 5 y 7.

35 Estos y otros aspectos, características y ventajas de la presente invención resultarán evidentes a partir de la siguiente descripción detallada de realizaciones a modo de ejemplo, que se debe leer en conexión con los dibujos adjuntos.

Breve descripción de los dibujos

La presente invención se puede comprender mejor de acuerdo con las siguientes figuras a modo de ejemplo, en las cuales:

40 la figura 1 es un diagrama de bloques que muestra un aparato de codificación de video a modo de ejemplo al que se pueden aplicar los presentes principios;

la figura 2 es un diagrama de bloques que muestra un aparato de decodificación de video a modo de ejemplo al que se pueden aplicar los presentes principios;

45 la figura 3 es un diagrama de flujo que muestra un proceso de codificación de video a modo de ejemplo con un bloque de transformada de color, de precodificación, de acuerdo con los presentes principios;

la figura 4 es un diagrama de flujo que muestra un proceso de decodificación de video a modo de ejemplo con un bloque de transformada de color inversa, de postdecodificación, de acuerdo con los presentes principios;

la figura 5 es un diagrama de bloques que muestra un modelo simplificado de transformada de color residual (RCT, residual color transform);

las figuras 6A y 6B son gráficos de PSNR promedio frente a velocidad binaria para ATV solamente intra, de acuerdo con los presentes principios;

las figuras 7A y 7B son gráficos de PSNR promedio frente a velocidad binaria para CT solamente intra, de acuerdo con los presentes principios;

5 las figuras 8A y 8B son gráficos de PSNR promedio frente a velocidad binaria para DT solamente intra, de acuerdo con los presentes principios;

las figuras 9A y 9B son gráficos de PSNR promedio frente a velocidad binaria para MIR_HD solamente intra, de acuerdo con los presentes principios;

10 las figuras 10A y 10B son gráficos de PSNR promedio frente a velocidad binaria para RT solamente intra, de acuerdo con los presentes principios;

las figuras 11A y 11B son gráficos de PSNR promedio frente a velocidad binaria para STB_HD solamente intra, de acuerdo con los presentes principios;

la figura 12 es una tabla que muestra sintaxis de parámetro de secuencia H.264, de acuerdo con los presentes principios;

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

la figura 14 es un diagrama de flujo que muestra un proceso de codificación de video a modo de ejemplo con un bloque de transformada de color, de precodificación, de acuerdo con los presentes principios;

20 la figura 15 es un diagrama de flujo que muestra un proceso de decodificación de video a modo de ejemplo, con un bloque de etapa de transformada de color inversa, de postdecodificación, de acuerdo con los presentes principios; y

la figura 16 es una tabla que muestra sintaxis de predicción de macrobloques H.264, de acuerdo con los presentes principios.

Descripción detallada

25 La presente invención está dirigida procedimientos y aparatos para codificación y decodificación de video de datos de señal de video. Se debe apreciar que aunque la presente invención se describe principalmente con respecto a datos de señal de video muestreados utilizando el formato 4:4:4 del estándar H.264 de la Unión Internacional de Telecomunicaciones, sector de telecomunicaciones (ITU-T), la presente invención se puede aplicar asimismo a datos de señal de video muestreados utilizando otros formatos (por ejemplo, el formato 4:2:2 y/o 4:2:0) del estándar H.264, así como otros estándares de compresión de video, manteniéndose el alcance de la presente invención.

30 Se debe apreciar que los procedimientos y aparatos acordes con los presentes principios no requieren la utilización de una o varias herramientas nuevas para el algoritmo de compresión luma o croma. En su lugar, se pueden utilizar las herramientas existentes de codificación luma. Por consiguiente, un resultado ventajoso de lo anterior es que el rendimiento de codificación del formato 4:4:4 se puede maximizar, manteniendo al mismo tiempo la compatibilidad hacia atrás y minimizando los cambios sobre el estándar H.264 existente (u otro aplicable).

35 De acuerdo con los principios de la presente invención configurada en una realización, se utiliza un algoritmo de codificación luma para codificar la totalidad de los tres canales componentes de, por ejemplo, contenido 4:4:4. Las ventajas de esta realización incluyen una mejora en el rendimiento general de la codificación para comprimir contenido 4:4:4 con respecto a la técnica anterior. Actualmente, en el estándar H.264 existente, solamente uno de los tres canales se codifica como luma, y los otros dos canales se codifican como croma utilizando herramientas menos eficientes.

40 Además, de acuerdo con los principios de la presente invención tal como se configura en una realización, se lleva a cabo transformación de color como una etapa de preprocesamiento. Por lo tanto, de acuerdo con esta realización, no se lleva a cabo una transformada de color residual (RCT) dentro del bucle de compresión. Las ventajas de esta realización incluyen la disposición de una arquitectura consistente de codificador/decodificador entre todos los formatos de color.

45 Asimismo, de acuerdo con una realización, en lugar de utilizar el mismo predictor para la totalidad de los tres componentes, se puede utilizar para los tres componentes un conjunto (o subconjunto) de tres (3) predictores espaciales restringidos. Las ventajas de esta realización incluyen una mejora del rendimiento global de la codificación para comprimir contenido 4:4:4, con respecto a la técnica anterior.

50 Se debe apreciar que las diversas realizaciones descritas en el presente documento anteriormente y en lo que sigue se pueden implementar como realizaciones independientes o se pueden combinar de cualquier modo, tal como apreciarán fácilmente los expertos en esta materia y otras relacionadas. De este modo, por ejemplo, en una primera

- realización combinada, un algoritmo de codificación luma se utiliza ventajosamente para codificar la totalidad de los tres canales componentes, se lleva a cabo transformación de color como una etapa de precodificación y se utiliza un único predictor para la totalidad de los tres canales componentes. En una segunda realización combinada, se utiliza ventajosamente un algoritmo de codificación luma para codificar la totalidad de los tres canales componentes, se
- 5 lleva a cabo transformación de color como una etapa de preprocesamiento y se puede utilizar un conjunto (o subconjunto) de tres (3) predictores espaciales restringidos, para los tres canales componentes. Por supuesto, tal como se ha indicado anteriormente, se pueden implementar asimismo otras combinaciones de las diversas realizaciones a partir de las explicaciones de los presentes principios dados a conocer la presente memoria, manteniéndose al mismo tiempo el alcance de la presente invención.
- 10 Todos los ejemplos y los términos condicionales enunciados en la presente memoria están destinados a fines pedagógicos para ayudar al lector en la comprensión de los principios de la invención y los conceptos con los que el inventor contribuye a desarrollar la técnica, y no se debe considerar que se limitan a dichos ejemplos y condiciones enunciados específicamente.
- 15 De este modo, por ejemplo, los expertos en la materia apreciarán que los diagramas de bloques presentados en la presente memoria representan vistas conceptuales de circuitos ilustrativos que realizan los principios de la invención. Asimismo, se apreciará que cualesquiera gráficos de flujo, diagramas de flujo, diagramas de transición de estado, pseudocódigo y similares representan diversos procesos que se pueden representar sustancialmente en medios legibles por ordenador y, asimismo, ser ejecutados por un ordenador o procesador, se muestre o no explícitamente dicho ordenador o procesador.
- 20 Las funciones de los diversos elementos mostrados en las figuras se pueden proporcionar mediante la utilización de hardware dedicado así como de hardware que puede ejecutar software, en asociación con el software apropiado. Cuando son proporcionadas por un procesador, las funciones se pueden proporcionar mediante un único procesador dedicado, mediante un único procesador compartido o mediante una serie de procesadores individuales, de los que algunos podrían ser compartidos. Además, no se deberá considerar que la utilización explícita del término
- 25 "procesador" o "controlador" se refiere exclusivamente a hardware capaz de ejecutar software, y puede incluir implícitamente, sin limitación, hardware de procesador de señal digital (DSP, digital signal processor), memoria de sólo lectura (ROM, hardware, read-only memory) para almacenar software, memoria de acceso aleatorio (RAM, random access memory) y almacenamiento no volátil.
- 30 Se puede incluir asimismo otro hardware, convencional y/o personalizado. De igual modo, cualesquiera conmutadores mostrados en las figuras son solamente conceptuales. Su función puede ser llevada a cabo por medio de la operación de lógica de programa, por medio de lógica dedicada, por medio de la interacción de lógica de control de programas y lógica dedicada, o incluso manualmente, pudiendo la técnica particular ser seleccionada por el implementador según su comprensión más específica del contexto.
- 35 En las reivindicaciones, cualquier elemento expresado como un medio para llevar a una función especificada pretende abarcar cualquier modo de llevar a cabo dicha función incluyendo, por ejemplo, a) una combinación de elementos de circuito que lleva a cabo dicha función o b) software en cualquier forma incluyendo, por lo tanto, software inalterable, microcódigo o similares, combinado con circuitos apropiados para ejecutar dicho software para la realización de la función. La invención, tal como se define mediante dichas reivindicaciones, reside en el hecho de que las funcionalidades dadas a conocer por los diversos medios enunciados se combinan y reúnen en el modo
- 40 requerido por las reivindicaciones.
- Pasando a la figura 1, un aparato de codificación de video a modo de ejemplo se indica, en general, por el numeral de referencia 199. El aparato de codificación de video 199 incluye un codificador de video 100 y un módulo de transformada de color de precodificación 105.
- 45 El módulo de transformada de color de precodificación 105 es para llevar a cabo el preprocesamiento de color de señales de video antes de introducir las en el codificador de video 100. El preprocesamiento de color realizado por el módulo de transformada de color de precodificación 105 se describe en mayor detalle a continuación. Se debe apreciar que el módulo de transformada de color de precodificación 105 se puede omitir en algunas realizaciones.
- Una entrada del módulo de transformada de color de precodificación 105 y una entrada del codificador de video 100 están disponibles como entrada del aparato de codificación de video 199.
- 50 Una salida del módulo de transformada de color de precodificación 105 está conectada en comunicación de señal con la entrada del codificador de video 100.
- La entrada del codificador de video 100 está conectada en comunicación de señal con una entrada sin inversión de un nodo de suma 110. La salida del nodo de suma 110 está conectada en comunicación de señal con un transformador/cuantificador 120. La salida del transformador/cuantificador 120 está conectada, en comunicación de
- 55 señal, con un codificador entrópico 140. Una salida del codificador entrópico 140 está disponible como una salida del codificador de video 100, y asimismo como una salida del aparato de codificación de video 199.

La salida del transformador/cuantificador 120 está conectada además en comunicación de señal con un transformador/cuantificador inverso 150. Una salida del transformador/cuantificador inverso 150 está conectada en comunicación de señal con una entrada del filtro anti efecto bloque 160. La salida del filtro anti efecto bloque 160 está conectada en comunicación de señal con almacenamientos de imágenes de referencia 170. Una primera salida de los almacenamientos de imágenes de referencia 170 está conectada en comunicación de señal con una primera entrada de un estimador de predicción de movimiento y espacial 180. La entrada al codificador de video 100 está conectada además en comunicación de señal con una segunda entrada del estimador de predicción de movimiento y espacial 180. La salida del estimador de predicción de movimiento y espacial 180 está conectada en comunicación de señal con una primera entrada de un compensador de predicción de movimiento y espacial 190. Una segunda salida de los almacenamientos de imágenes de referencia 170 está conectada en comunicación de señal con una segunda entrada del compensador de movimiento y espacial 190. La salida del compensador de movimiento y espacial 190 está conectada en comunicación de señal con una entrada de inversión del nodo de suma 110.

Pasando a la figura 2, un aparato de descodificación de video a modo de ejemplo se indica, en general, por el numeral de referencia 299. El aparato de descodificación de video 299 incluye un descodificador de video 200 y un módulo de transformada de color inversa de postdescodificación 293.

Una entrada del descodificador de video 200 está disponible como una entrada del aparato de descodificación de video 299. La entrada al descodificador de video 200 está conectada, en comunicación de señal, con una entrada del descodificador entrópico 210. Una primera salida del descodificador entrópico 210 está conectada, en comunicación de señal, con una entrada del cuantificador/transformador inverso 220. Una salida del cuantificador/transformador inverso 220 está conectada, en comunicación de señal, con una primera entrada de un nodo de suma 240.

La salida del nodo de suma 240 está conectada en comunicación de señal con un filtro anti efecto bloque 290. Una salida del filtro anti efecto bloque 290 está conectada en comunicación de señal con almacenamientos de imágenes de referencia 250. El almacenamiento de imágenes de referencia 250 está conectado en comunicación de señal con una primera entrada de un compensador de predicción de movimiento y espacial 260. Una salida del compensador de predicción espacial de movimiento 260 está conectada, en comunicación de señal, con una segunda entrada del nodo de suma 240. Una segunda salida del descodificador entrópico 210 está conectada, en comunicación de señal, con una segunda entrada del compensador de movimiento 260. La salida del filtro anti efecto bloque 290 está disponible como una salida del descodificador de video 200 y, asimismo, como una salida del aparato de descodificación de video 299.

Además, una salida del módulo de transformada de color inversa, de postdescodificación 293 puede estar disponible como una salida del aparato de descodificación de video 299. En tal caso, la salida del descodificador de video 200 puede estar conectada en comunicación de señal con una entrada del módulo de transformada de color inversa, de postdescodificación 293, que es un módulo de postprocesamiento con respecto al descodificador de video 200. Una salida del módulo de transformada de color inversa, de postdescodificación 293 proporciona una señal transformada de color inversa, postprocesada, con respecto a la salida del descodificador de video 200. Se debe apreciar asimismo que la utilización del módulo de transformada de color inversa, de postdescodificación 293 es opcional.

Se presenta a continuación una descripción para una codificación 4:4:4 mejorada. Un primer ejemplo de antecedentes ilustrativo descrito es una realización combinada en la que se utiliza el algoritmo de codificación luma para todos los componentes de color, se utiliza el mismo modo de predicción espacial para todos los componentes de color y la transformada de color residual (RCT) se omite del bucle de compresión. Se proporcionan asimismo resultados de pruebas para esta realización combinada. Después de esto, se describe una realización combinada en la que el algoritmo de codificación luma se utiliza para todos los componentes de color, un conjunto (o subconjunto) de predictores espaciales restringidos se utiliza para todos los componentes de color (en lugar de un único modo de predicción espacial) y la transformada de color residual (RCT) se omite dentro del bucle de compresión. Por lo tanto, la diferencia entre el ejemplo de antecedentes y la realización es la utilización de un único modo de predicción espacial para todos los componentes de color en el ejemplo del antecedente, frente a la utilización de un conjunto (o subconjunto) de predictores espaciales restringidos, para todos los componentes de color en la realización. Por supuesto, tal como se ha indicado anteriormente, las realizaciones descritas en la presente memoria se pueden implementar como realizaciones independientes o se pueden combinar de cualquier modo, tal como apreciará fácilmente un experto en esta materia y otras relacionadas. Se debe apreciar que, dadas las explicaciones de los presentes principios dados a conocer aquí, estas y otras variaciones, implementaciones y combinaciones de las realizaciones de la presente invención serán fácilmente determinables por un experto en esta materia y otras relacionadas, manteniéndose al mismo tiempo el alcance de la presente invención.

Volviendo a la figura 3, un proceso de codificación de video a modo de ejemplo con un bloque de transformada de color de precodificación, se indican, en general, por los numerales de referencia 300 y 301, respectivamente.

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

El bloque de transformada de color de precodificación 301 incluye un bloque de límite de bucle 306 que comienza un bucle para cada bloque en una imagen, y pasa el control a un bloque de función 308. El bloque de función 308 lleva a cabo preprocesamiento de color de los datos de señal de video del bloque de imagen actual, y pasa el control a un bloque de límite de bucle 310. El bloque de límite de bucle 310 finaliza el bucle. Además, el bloque de límite de bucle 310 pasa el control a un bloque de límite de bucle 312, estando este último incluido en el proceso de codificación de video 300.

El bloque de límite de bucle 312 comienza un bucle para cada bloque en la imagen, y pasa el control a un bloque de función 315. El bloque de función 315 forma una predicción espacial o compensada en movimiento del bloque de imagen actual, utilizando un predictor común para cada componente de color del bloque de imagen actual, y pasa el control a un bloque de función 320. El bloque de función 320 resta la predicción espacial o compensada en movimiento del bloque de imagen actual, para formar un residuo de predicción, y pasa el control al bloque de función 330. El bloque de función 330 transforma y cuantifica el residuo de predicción, y pasa el control a un bloque de función 335. El bloque de función 335 realiza la transformada y cuantificación inversa del residuo de predicción para formar un residuo de predicción codificado, y pasa el control a un bloque de función 345. El bloque de función 345 añade el residuo codificado a la predicción, para formar un bloque de imagen codificado, y pasa el control a un bloque de fin de bucle 350. El bloque de fin de bucle 350 finaliza el bucle y pasa el control a bloque final 355.

Pasando a la figura 4, un proceso de descodificación de video a modo de ejemplo, con un bloque de transformada de color inversa, de postdescodificación, se indican generalmente mediante los numerales de referencia 400 y 460, respectivamente

Se debe apreciar que el bloque de transformada de color inversa, de postdescodificación 460 incluye los bloques 462, 464, 466 y 468. Además, se debe apreciar que el bloque de transformada de color inversa, de postdescodificación 460 es opcional y, por lo tanto, se puede omitir en algunas realizaciones de la presente invención.

El proceso de descodificación 400 incluye un bloque de límite de bucle 410 que comienza un bucle para un bloque actual en una imagen, y pasa el control a un bloque de función 415. El bloque de función 415 realiza la descodificación entrópica del residuo codificado, y pasa el control a un bloque de función 420. El bloque de función 420 realiza la transformada y cuantificación inversa del residuo descodificado para formar un residuo codificado, y pasa el control a un bloque de función 430. El bloque de función 430 añade el residuo codificado a la predicción formada a partir de un predictor común para cada componente de color, para formar un bloque de imagen codificada, y pasa el control al bloque de límite de bucle 435. El bloque de límite de bucle 435 finaliza el bucle y pasa el control a bloque final 440.

En algunas realizaciones, el bloque de límite de bucle 435 pasa opcionalmente el control al bloque de transformada de color inversa, de postdescodificación 460, en particular, al bloque de límite de bucle 462 incluido en el bloque de transformada de color inversa, de postdescodificación 460. El bloque de límite de bucle 462 comienza un bucle para cada bloque en una imagen, y pasa el control a un bloque de función 464. El bloque de función 464 lleva a cabo un postprocesamiento de color inverso de los datos de señal de video del bloque de imagen actual, y pasa el control a un bloque de límite de bucle 466. El bloque de límite de bucle 466 finaliza el bucle, y pasa el control a un bloque final 468.

En el formato 4:4:4 H.264, cada canal de componente tiene plena resolución. Por lo tanto, según el ejemplo de antecedentes expuesto anteriormente, el algoritmo de codificación luma se utiliza en cada componente de color para conseguir la máxima eficiencia de compresión global. Por consiguiente, en la realización, para cuadros intra, cada componente de color se puede comprimir, por ejemplo, utilizando los modos de predicción enumerados en la tabla 8-2, la tabla 8-3 y la tabla 8-4 en ISO/IEC 14496 10 Advanced Video Coding 3rd Edition (ITU-T Rec. H.264), ISO/IEC JTC1/SC29/WG11 e ITU-T SG16 Q.6, documento N6540, julio de 2004.

Además, se utiliza el mismo modo de predicción espacial para la totalidad de los tres componentes de píxel, para reducir más la complejidad del códec y mejorar el rendimiento. Por ejemplo, el modo de predicción establecido por los parámetros `prev_intra4x4_pred_mode_flag`, `rem_intra4x4_pred_mode`, `prev_intra8x8_pred_mode_flag` y `rem_intra8x8_pred_mode` para luma en la cabecera de predicción de macrobloques puede ser utilizado por la totalidad de los tres componentes. Por lo tanto, no se requieren elementos de sintaxis y bits extra. Para los cuadros B y P (predictivo), los píxeles de referencia en posiciones de píxeles fraccionarias se pueden calcular mediante los procedimientos de interpolación descritos en la sección 8.4.2.2.1 del estándar H.264 para la totalidad de los tres canales. Los cambios de sintaxis y semántica detallados en el estándar H.264 actual se explicarán en mayor detalle a continuación en este documento.

Se ha añadido la transformada de color residual (RCT) al codificador/descodificador en el perfil 4:4:4 alto. Como resultado, la estructura de compresión para el formato 4:4:4 es diferente de la utilizada actualmente en la totalidad de los otros perfiles en el estándar H.264 para formatos 4:2:0 y 4:2:2. Esto tiene como resultado alguna complejidad extra para la implementación. Además, de manera similar a cualesquiera otras transformadas de color, YCOCG no siempre mejora el rendimiento de compresión global. La eficacia de YCOCG es extremadamente dependiente del contenido. Por lo tanto, para mejorar la compresión y robustez globales, la transformada de color se sitúa fuera del bucle de predicción, como parte del bloque de preprocesamiento. Al hacerlo así, seleccionar una transformada de

5 color óptima para una tarea de compresión específica es una cuestión operativa, y la mejor respuesta para una secuencia de entrada particular se encontraría entre una serie de opciones. De acuerdo con un ejemplo de antecedentes en el que la totalidad de los tres componentes utilizan los mismos predictores espaciales para los cuadros intra y los mismos filtros de interpolación para los cuadros B y P (predictivos o codificados inter), que se lleve a cabo la transformada de color sobre los residuos de predicción es idéntico a llevar a cabo la transformada de color sobre las imágenes de origen fuera del códec cuando se ignoran los errores de redondeo/truncado. Esto se explicará aquí en mayor detalle a continuación. Por lo tanto, el bloque RCT se elimina de la estructura de codificación para hacer consistente la estructura de codificación entre todos los formatos de color.

10 Volviendo a la figura 5, se indica un modelo simplificado de RCT, en general, mediante el numeral de referencia 500. El modelo RCT 500 incluye un generador de píxeles de referencia 510, un nodo de suma 520 y un módulo de transformada lineal 530. Las entradas al generador de píxeles de referencia 510 se configuran para recibir información de movimiento/bordes y vectores $[X_1], [X_2], \dots, [X_n]$. Una salida del generador de píxeles de referencia 510 está conectada en comunicación de señal con una entrada de inversión del nodo de suma 520, que proporciona un vector de predicción $[X_p]$ a la misma. Una entrada sin inversión del nodo de suma 520 está configurada para recibir un vector de entrada $[X_{in}]$. Una salida del nodo de suma 520 está conectada en comunicación de señal con una entrada del módulo de transformada lineal 530, que proporciona un vector $[X_d]$ a la misma. Una salida del módulo de transformada lineal 530 está configurada para proporcionar un vector $[Y_d]$.

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

$$20 \quad \begin{bmatrix} Y \\ u \\ v \end{bmatrix} = [A] \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

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. $[Y_d]$ es un vector 3×1 que representa el resultado de la transformada de color. Por lo tanto,

$$[Y_d] = [A][X_d] = [A][X_{in}] - [A][X_p] \quad (2)$$

25 Dado que se utilizan los mismos predictores espaciales y filtros de interpolación para la totalidad de los tres componentes en un macrobloque, el píxel de referencia $[X_p]$ se puede expresar como sigue:

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

donde un vector $n \times 1$ $[C]$ representa las operaciones lineales involucradas en los predictores espaciales y los filtros de interpolación definidos en el estándar H.264. En este caso, se supone que el píxel de referencia se calcula utilizando un número total de n píxeles vecinos $[X_1], [X_2], \dots, [X_n]$.

30 Sustituir $[X_p]$ de la ecuación (3) en la ecuación (2) tiene como resultado lo siguiente:

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

Ignorar los errores de redondeo/truncado y suponer que se selecciona el mismo modo de predicción en los dominios RGB o Y tiene como resultado lo siguiente:

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

Por lo tanto,

$$[Y_d] = [Y_{in}] - [Y_1 Y_2 Y_3 \dots Y_n] \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix} \quad (6)$$

5 De ese modo, la ecuación (6) muestra claramente que utilizar YUV como la entrada al codificador/descodificador es idéntico a llevar a cabo RCT.

Asimismo, se añade un nuevo perfil 4:4:4 al estándar H.264, denominado en la presente memoria "perfil 4:4:4 avanzado con profile_idc=166". Este nuevo profile_idc se puede añadir en la cabecera de parámetros de secuencia, y se puede utilizar en la cabecera de capas de macrobloques, así como en la cabecera de datos residuales.

10 Para soportar la utilización del algoritmo luma para codificar la totalidad de los tres componentes de color, se pueden realizar algunos cambios a la sintaxis de datos residuales. Además, se pueden realizar cambios asimismo a la semántica de algunos de los elementos de la cabecera de macrobloques, la cabecera de datos residuales, etc. En general, la sintaxis existente para luma en la especificación H.264 permanecerá sin cambios y se utilizará para codificar uno de los tres componentes. Los cambios son compatibles hacia atrás. Se describen a continuación los cambios detallados de sintaxis y semántica.

15 Se proporcionará una descripción en relación con resultados de simulación.

Pasando a las figuras 6A y 6B, los gráficos de PSNR promedio frente a velocidad binaria para ATV solamente intra se indican, en general, mediante los numerales de referencia 600 y 650, respectivamente.

Pasando a las figuras 7A y 7B, los gráficos de PSNR promedio frente a velocidad binaria para CT solamente intra se indican en general mediante los números numerales de referencia 700 y 750, respectivamente.

20 Pasando a las figuras 8A y 8B, los gráficos de PSNR promedio frente a velocidad binaria para DT solamente intra se indican en general mediante los numerales de referencia 800 y 850.

Pasando a las figuras 9A y 9B, los gráficos de PSNR promedio frente a velocidad binaria para MIR_HD solamente intra se indican en general por los numerales de referencia 900 y 950, respectivamente.

25 Pasando a las figuras 10A y 10B, los gráficos de PSNR promedio frente a velocidad binaria para RT solamente intra se indican en general por los numerales de referencia 1000 y 1050, respectivamente.

Pasando a las figuras 11A y 11B, los gráficos de PSNR promedio frente a velocidad binaria para STB_HD solamente intra se indican en general por los numerales de referencia 1100 y 1150.

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

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

35 Todas las secuencias de prueba JVT/FRExt descritas en secuencias de prueba originadas en película, JVT-J042, se utilizaron en las pruebas. Son todas ellas material de película de 10 bits 4:4:4 y cada clip tiene 58 cuadros.

Los perfiles 4:4:4 avanzados propuestos se implementaron en el software de referencia JVT JM9.6. En las pruebas se utilizó una estructura de codificación tanto intra solamente como IBBP. El parámetro de cuantificación se ajustó a 6, 12, 18, 24, 30 y 42 para cada una de las curvas R-D. Se utilizó selección de modo optimizado en RD.

El perfil 4:4:4 avanzado propuesto se comparó asimismo con los resultados que se obtuvieron ejecutando software de referencia con el YUVFormat=0(4:0:0) en cada componente de entrada individual. Se añadieron conjuntamente simplemente tres cómputos de bits comprimidos individuales separados, para obtener los bits comprimidos totales para la velocidad binaria comprimida

5 En relación con JPEG2k, en las pruebas se utilizó un software KaKadu V2.2.3. Los resultados de las pruebas se generaron utilizando 5 niveles de descompresión de ondículas con filtro de ondículas biortogonal de 9/7 parámetros. Había solamente un título por cuadro y se utilizó asimismo RDOptimización para una determinada velocidad objetivo.

10 Todas las mediciones de PSNR se realizaron en el dominio RGB. Se utilizó la PSNR promedio, definida como $(PSNR(rojo)+PSNR(verde)+PSNR(azul))/3$, para comparar la calidad de compresión global. Esto se debe principalmente a que los datos comprimidos JPEG2k se calcularon utilizando un algoritmo de control de velocidad desconocido, proporcionado por el software. Para algunos casos, los valores PSNR de RGB están muy separados entre sí, especialmente cuando se ha utilizado transformada de color JPEG2k.

La comparación de compresión se llevó a cabo como sigue:

- 15
- New1: el perfil 4:4:4 avanzado propuesto, con un único modo de predicción.
 - New3: el perfil 4:4:4 avanzado propuesto, con tres modos de predicción.
 - RCT-OFF: entrada RGB con RCT=off.
 - RCT-ON: entrada RGB con RCT=on.
- 20
- YCOCG: la conversión RGB a YCOCG se realizó fuera del códec. A continuación se utilizó YCOCG transformado como la entrada al software JVT.
 - R+G+B: procedimiento propuesto, aproximado mediante compresión de las señales R, G y B por separado.
 - Y+CO+CG: procedimiento propuesto, aproximado comprimiendo por separado las señales Y, CO y CG transformadas.
 - J2k_RGB: la compresión JPEG2k se realizó en el dominio RGB. La transformada de color JPEG2k se desactivó.
- 25
- J2k_YUV: la compresión JPEG2k se realizó en el dominio YUV. Se utilizó la transformada de color JPEG2k.

De acuerdo con los resultados de las pruebas, una implementación en general, es muy similar a JPEG2k en términos de eficiencia de compresión global. En algunos casos, es incluso algo mejor.

30 Además, una implementación proporciona un rendimiento significativamente mayor (compresión) que el actual perfil 4:4:4 alto para calidad por encima de 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 mayor que 1,5 dB. En el último ejemplo, la mejora se puede traducir a un ahorro de bits de más del 25 % a una PSNR igual a 45 dB.

35 De acuerdo con los resultados de las pruebas, parece que las transformadas de color contribuirán al rendimiento de la codificación cuando el contenido está más saturado en color, tal como TP, RT. Es decir, si el color es neutro y está menos saturado, codificar en el dominio RGB puede ser la mejor elección. La observación anterior es independiente de qué transformada de color se utilice.

40 Comparando los resultados de New1-YCOCG o New3-YCOCG y JPEG-2k_YUV, se ha observado que el rendimiento de una transformada de color específica en términos de mejorar la eficiencia de codificación es muy dependiente del contenido. Ninguna única transformada de color es siempre la mejor. Por lo tanto, los datos de los inventores confirmaron que tener una transformada de color, tal como RCT, dentro del bucle de codificación (o decodificación) puede no ser una buena idea. En cambio, realizar la transformada de color, si es necesaria, fuera del codificador/decodificador haría que todo el sistema de compresión proporcionara un rendimiento mejor y más robusto.

45 Comparando YCOCG con RCT-ON, los resultados de las pruebas no muestran ninguna mejora en la eficiencia de codificación frente a RCT. Además, se debe observar que ejecutar el software de referencia con la RCT activada aumentó significativamente el tiempo de codificación. El tiempo de ejecución fue más de 2,5 veces más largo.

Se proporcionará a continuación una descripción en relación con cambios de sintaxis y semántica.

Volviendo a la figura 12, una tabla para sintaxis de parámetros de secuencia H.264 se indica, en general, por el numeral de referencia 1200. Los cambios en la sintaxis se indican mediante texto en cursiva.

Pasando a la figura 13, una tabla para sintaxis de datos residuales H.264 se indica, en general, por el numeral de referencia 1300. Las adiciones/cambios en la sintaxis se indican mediante texto en cursiva. En la tabla 1300, una sección de luma en la cabecera de datos residuales junto con algunas modificaciones de texto necesarias se repite dos veces para soportar respectivamente luma1 y luma2.

- 5 Tal como se ha indicado anteriormente, el ejemplo de antecedentes descrito anteriormente fue evaluado y probado implementando los presentes principios en el software de referencia JVT JM9.6. Los resultados de las pruebas marcados con New1-RGB o New1-YCOCG representan el ejemplo de antecedentes.

- 10 Tal como se ha indicado anteriormente, de acuerdo con los principios de la presente invención tal como se configura en una realización, un conjunto (o subconjunto) de tres (3) predictores espaciales restringidos se utiliza para los canales componentes (por ejemplo, formatos RGB, YUV, YCrCb, etc.) en lugar de un único modo de predicción espacial. Además, tal como se ha indicado anteriormente, esta realización se puede combinar con otras realizaciones descritas en el presente documento, tales como, por ejemplo, la utilización de solamente el algoritmo de codificación luma para codificar la totalidad de los tres canales componentes de contenido y/o la utilización de transformación de color como una etapa de preprocesamiento.

- 15 A continuación se proporcionará una descripción en relación con la realización combinada descrita anteriormente, que involucra la utilización de un conjunto (o subconjunto) de tres (3) predictores espaciales restringidos para los componentes de color, la utilización de solamente el algoritmo de codificación luma para codificar la totalidad de los tres componentes de color y la utilización de transformación de color como una etapa de preprocesamiento (es decir, sin RCT dentro del bucle de compresión). También se describirán con esto algunas variaciones de esta realización.

- 20 Pasando a la figura 14, un proceso de codificación de video a modo de ejemplo con una etapa de transformada de color de precodificación, se indican en general por los numerales de referencia 1400 y 1401, respectivamente.

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

- 25 El bloque de transformada de color de precodificación 1401 incluye un bloque de límite de bucle 1406 que comienza un bucle para cada bloque en una imagen, y pasa el control a un bloque de función 1408. El bloque de función 1408 lleva a cabo preprocesamiento de color de los datos de señal de video del bloque de imagen actual, y pasa el control a un bloque de límite de bucle 1410. El bloque de límite de bucle 1410 finaliza el bucle. Además, el bloque de límite de bucle 1410 pasa el control a un bloque de límite de bucle 1412, estando este último incluido en el proceso de codificación de video 1400.

- 30 El bloque de límite de bucle 1412 comienza un bucle para cada bloque en la imagen, y pasa el control a un bloque de función 1415. El bloque de función 1415 forma una predicción espacial o compensada en movimiento del bloque de imagen actual, utilizando un predictor común para cada componente de color del bloque de imagen actual, y pasa el control a un bloque de función 1420. El bloque de función 1420 resta la predicción espacial o compensada en movimiento del bloque de imagen actual, para formar un residuo de predicción, y pasa el control a un bloque de función 1430. El bloque de función 1430 transforma y cuantifica el residuo de predicción, y pasa el control a un bloque de función 1435. El bloque de función 1435 realiza la transformada y cuantificación inversa del residuo de predicción para formar un residuo de predicción codificado, y pasa el control a un bloque de función 1445. El bloque de función 1445 añade el residuo codificado a la predicción, para formar un bloque de imagen codificado, y pasa el control a un bloque de fin de bucle 1450. El bloque de fin de bucle 1450 finaliza el bucle y pasa el control a un bloque final 1455.

Pasando a la figura 15, un proceso de descodificación de video a modo de ejemplo con una etapa de transformada de color inversa, de postdescodificación, se indican en general mediante los numerales de referencia 1500 y 1560, respectivamente.

- 45 Se debe apreciar que el bloque de transformada de color inversa, de postdescodificación 1560 incluye bloques 1562, 1564, 1566 y 1568. Además, se debe apreciar que el bloque de transformada de color inversa, de postdescodificación 1560 es opcional y, por lo tanto, se puede omitir en algunas realizaciones de la presente invención.

- 50 El proceso de descodificación 1500 incluye un bloque de límite de bucle 1510 que comienza un bucle para un bloque actual en una imagen, y pasa el control a un bloque de función 1515. El bloque de función 1515 realiza la descodificación entrópica del residuo codificado, y pasa el control a un bloque de función 1520. El bloque de función 1520 realiza la transformada y cuantificación inversa del residuo descodificado para formar un residuo codificado, y pasa el control a un bloque de función 1530. El bloque de función 1530 añade el residuo codificado a la predicción formada a partir de un predictor común para cada componente de color, para formar un bloque de imagen codificada, y pasa el control a un bloque de límite de bucle 1535. El bloque de límite de bucle 1535 finaliza el bucle, y pasa el control a un bloque final 1540.

En algunas realizaciones, el bloque de límite de bucle 1535 pasa opcionalmente el control al bloque de transformada de color inversa, de postdescodificación 1560, en particular, al bloque de límite de bucle 1562 incluido en el bloque de transformada de color inversa, de postdescodificación 1560. El bloque de límite de bucle 1562 comienza un bucle para cada bloque en una imagen, y pasa el control a un bloque de función 1564. El bloque de función 1564 lleva a cabo un postprocesamiento de color inverso de los datos de señal de video del bloque de imagen actual, y pasa el control a un bloque de límite de bucle 1566. El bloque de límite de bucle 1566 finaliza el bucle, y pasa el control a un bloque final 1568.

Tal como se ha indicado anteriormente, se da a conocer un nuevo perfil (`profile_idc=166`) para el perfil 4:4:4 avanzado. Este nuevo perfil se puede utilizar asimismo para la realización combinada, con los correspondientes cambios semánticos y sintácticos que se describen en este documento a continuación para la realización combinada. Este nuevo `profile_idc` se añade en el conjunto de parámetros de secuencia y se utiliza principalmente en las subsiguientes cabeceras para indicar que el formato de entrada es 4:4:4 y la totalidad de los tres canales de entrada se codifican de manera similar como luma.

Para minimizar los cambios necesarios en el estándar H.264, no se da a conocer ningún nuevo tipo de macrobloque para el perfil 4:4:4 avanzado. En cambio, siguen siendo válidos todos los tipos de macrobloques junto con los parámetros de codificación asociados enumerados en la tabla 7-11, la tabla 7-13 y la tabla 7-14 del estándar H.264. Para el caso de macrobloques intra, tres canales de entrada, luma, Cr y Cb, se codificarán todos en base al `MbPartPredMode` definido en la tabla 7-11 del estándar H.264. Por ejemplo, un macrobloque `Intra_4x4` en el perfil 4:4:4 avanzado significa que cada canal componente de entrada se puede codificar utilizando la totalidad de los 9 posibles modos de predicción proporcionados en la tabla 8-2 del estándar H.264. Como referencia, en el actual perfil 4:4:4 alto, dos de los canales para un macrobloque `Intra_4x4` se tratarán como croma y solamente se utilizará uno de los 4 posibles modos de predicción intra en la tabla 8-5 del estándar H.264. Para los macrobloques B y P, los cambios realizados para el perfil 4:4:4 avanzado ocurren en el proceso de interpolación para el cálculo del valor de píxel de referencia en la posición del píxel fraccionario. En este caso, el procedimiento descrito en la sección 8.4.2.2.1 del estándar H.264, procesador de interpolación de muestras luma, se aplicará para luma, Cr y Cb. De nuevo como referencia, el actual perfil 4:4:4 alto utiliza la sección 8.4.2.2.2 del estándar H.264, proceso de interpolación de muestras croma, para dos de los canales de entrada.

En el caso en que se elige CABAC como el modo de codificación entrópica, dos conjuntos independientes de modelos de contexto idénticos a los definidos actualmente para luma se crearán para Cr y Cb. Asimismo, se actualizarán independientemente en el curso de la codificación.

Finalmente, en la realización, dado que no hay ningún bloque RCT en el bucle de codificación, el `ResidueColorTransformFlag` se elimina del conjunto de parámetros de secuencia en el perfil 4:4:4 avanzado.

Hasta este momento, la mayor parte de los cambios sintácticos ocurren en los datos de residuo tal como se muestra en la figura 13, donde la sintaxis original para luma se repite dos veces para soportar Cr y Cb en los perfiles 4:4:4 avanzados propuestos.

En relación con la tabla de capas de macrobloque H.264 (no mostrada), los cambios semánticos a la sintaxis correspondiente incluyen los siguientes.

`coded_block_pattern` (añadir). Cuando `chroma_format_idc` es igual a 3 y `coded_block_pattern` está presente, `CodedBlockPatternChroma` se ajustará a 0. Además, `CodedBlockPatternLuma` específica, 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 transformada de los doce bloques luma 4x4 en los bloques luma 8x8, Cb 8x8 y Cr 8x8 son iguales a cero; (2) uno o varios niveles de coeficientes de transformada de uno o varios de los bloques luma 4x4 en los bloques luma 8x8, Cb 8x8 y Cr 8x8 deberán tener valores distintos de cero.

A continuación se proporcionará una descripción relacionada con la selección de modo de predicción espacial para los bloques intra, de acuerdo con la realización combinada (o la única realización relativa a la utilización del conjunto (o subconjunto) de tres predictores espaciales restringidos).

Para que cada componente elija independientemente su mejor `MbPartPredMode` y el subsiguiente mejor modo de predicción espacial, tal como en el caso cuando se codifica por separado cada canal de entrada, se pueden añadir algunos nuevos tipos de bloques intra a la tabla 7-11 del estándar H.264. Como resultado, se realizará una gran cantidad de cambios en el estándar H.264. En una realización relacionada con la realización combinada, los `mb_types` permanecen sin cambios y se da a conocer una solución alternativa. En la realización, los tres canales de entrada están restringidos a ser codificados con el mismo `MbPartPredMode` o tipo de macrobloque. A continuación, se añade una pequeña cantidad de nuevos elementos a la sintaxis de predicción de macrobloques para soportar tres modos de predicción independientes. Por lo tanto, cada componente puede teóricamente seguir eligiendo independientemente su mejor modo de predicción espacial, para minimizar el error de predicción para cada canal componente. Por ejemplo, suponiendo que se elige un macrobloque `Intra_4x4` como el `mb_type`, luma, Cr o Cb puede seguir encontrando su propio mejor modo de predicción espacial en la tabla 8-2, en la sección 8.3.1.1 del estándar 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.

- Otro enfoque, en relación con el ejemplo de antecedentes descrito anteriormente, es restringir la totalidad de los tres canales de entradas a compartir el mismo modo de predicción. Esto se puede realizar utilizando la información de predicción que transportan actualmente los elementos de sintaxis existentes, tales como `prev_intra4x4_pre_mode_flag`, `rem_intra4x4_pred_mode`, `pre_intra8x8_pred_mode_flag` y `rem_intra8x8_pred_mode`, en sintaxis la de predicción de macrobloques. Esta opción tendrá como resultado menos cambios en el estándar H.264 y asimismo una pérdida más leve de la eficiencia de codificación.
- Basándose en los resultados de las pruebas, utilizar tres modos de predicción mejoraría el rendimiento de codificación global en aproximadamente 0,2 dB sobre la primera realización combinada.
- 10 Pasando a la figura 16, una tabla para sintaxis de predicción de macrobloques H.264 se indica, en general, por el numeral de referencia 1700. Como referencia, la sintaxis modificada de predicción de macrobloques para soportar la utilización de los tres modos de predicción se enumera a continuación, donde:
- `prev_intra4x4_pred_mode_flag0` y `rem_intra4x4_pred_mode0` son para luma;
- `prev_intra4x4_pred_mode_flag1` y `rem_intra4x4_pred_mode1` son para Cr;
- `prev_intra4x4_pred_mode_flag2` y `rem_intra4x4_pred_mode2` son para Cb;
- 15 A continuación se proporcionará una descripción en relación con resultados de simulación obtenidos de acuerdo con los principios de la presente invención tal como se configura en una realización.
- Todos las JVT/FRExt prueban secuencias descritas en la secuencia de pruebas original en película, JVT-J042, JVT-J039 (Viper). Son todos materiales 4:4:4 de 10 bits, y cada clip tiene 58 cuadros.
- 20 El algoritmo propuesto se implementó en el software de referencia JVT JM9.6 y el software modificado se utilizó en las pruebas. Se probó tanto sólo intra como IBRRBP. En este caso, "Br" significa las imágenes B registradas. El caso solamente intra se realizó para la totalidad de 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 IBRRBP GOP se realizó solamente para los clips de película con parámetros de cuantificación igual a 12, 18, 24, 30 y 36. De acuerdo con la discusión en el 4:4:4 AHG, se utilizaron los siguientes parámetros clave en las pruebas:
- 25 `SymbolMode = 1`
`RDOptimization = 1`
`ScalingMatrixPresentFlag = 0`
`OffsetMatrixPresentFlag = 1`
`QoffsetMatrixFile = "q_offset.cfg"`
- 30 `AdaptiveRounding = 1`
`AdaptRndPeriod = 1`
`AdaptRndChroma = 1`
`AdaptRndWFactorX = 8`
`SearchRange = 64`
- 35 `UseFME = 1`
- Con respecto a JPEG2k, se utilizó un software KaKadu V2.2.3 en las pruebas. Los resultados de las pruebas se generaron utilizando 5 niveles de descompresión de ondículas con filtro de ondículas biortogonal de 9/7 parámetros. Había solamente un título por cuadro y se utilizó asimismo RDOptimización para una determinada velocidad objetivo.
- 40 Las mediciones de la PSNR se calcularon principalmente en el dominio de color original de los contenidos fuente, que es RGB para los clips descritos anteriormente. Se utilizó la PSNR promedio, definida como $(PSNR(rojo)+PSNR(verde)+PSNR(azul))/3$, para comparar la calidad de compresión global.
- La comparación de compresión se realizó como sigue:
- New1: el perfil 4:4:4 avanzado propuesto, con un único modo de predicción.
- 45 New3: el perfil 4:4:4 avanzado propuesto, con tres modos de predicción.
- RCT-OFF: entrada RGB con RCT=off.

RCT-ON: entrada RGB con RCT=on.

YCOCG: la conversión RGB a YCOCG se realizó fuera del códec. A continuación se utilizó YCOCG transformado como la entrada al software JVT.

R+G+B: procedimiento propuesto, aproximado mediante compresión de las señales R, G y B por separado.

- 5 Y+CO+CG: procedimiento propuesto, aproximado comprimiendo por separado las señales Y, CO y CG transformadas.

JPEG2k_RGB: la compresión JPEG2k se realizó en el dominio RGB. La transformada de color JPEG2k se desactivó.

JPEG2k_YUV: la compresión JPEG2k se realizó en el dominio YUV. Se utilizó la transformada de color JPEG2k.

- 10 Para el caso de solamente intra, el perfil 4:4:4 avanzado propuesto, de acuerdo con los presentes principios, es muy similar a JPEK2k en términos de eficiencia de compresión global. En algunos casos, es incluso algo mejor.

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

- 15 Incluso con el mismo tipo de bloque, utilizar tres modos de predicción es ligeramente mejor que solamente uno. Sin embargo, se pueden utilizar más cambios sintácticos y semánticos.

A continuación se proporcionará una descripción de algunas de las ventajas/características concomitantes, proporcionadas por los principios de realizaciones de la presente invención.

- 20 Los resultados de las pruebas demuestran que el perfil 4:4:4 avanzado propuesto, utilizando las mejoras correspondientes a los principios de la presente invención, proporciona un rendimiento mejorado en comparación con el actual perfil 4:4:4 alto. La ganancia de rendimiento es significativa. Además, desplazar la transformada de color fuera del códec hará consistente la arquitectura del códec entre todos los formatos de color. Como resultado, esto facilitará la implementación y reducirá el coste. Esto hará asimismo el códec más robusto en términos de seleccionar la transformada de color óptima para conseguir una mejor eficiencia de codificación. Asimismo, el enfoque propuesto no añade nuevas herramientas de codificación y requiere solamente algunos ligeros cambios en la sintaxis y la semántica.

- 25 Por lo tanto, de acuerdo con los principios de la presente invención tal como se configura en una realización, se da a conocer un procedimiento y un aparato para codificación y decodificación de video. Las modificaciones al estándar H.264 existentes se proporcionan con un rendimiento mejorado más allá del que se puede conseguir actualmente. Además, el rendimiento se mejora incluso más allá de JPEG-2000 para aplicaciones de alta calidad. Se pueden conseguir mejoras significativas de rendimiento de codificación 4:4:4 en el estándar H.264 utilizando el algoritmo de codificación luma para codificar la totalidad de los tres componentes de color de contenido 4:4:4. Es decir, no son necesarias herramientas nuevas para el algoritmo de compresión luma (o croma, que no se utiliza). En su lugar, se utilizan las herramientas de codificación luma existentes. Además, se pueden implementar cambios sintácticos y semánticos al perfil 4:4:4 actual, para soportar la codificación luma de la totalidad de los tres canales componentes. En las pruebas llevadas a cabo, cuando el contenido fuente tiene pérdidas de texturas espaciales y bordes, las herramientas de predicción espacial utilizadas en luma presentan claramente su rendimiento superior a las utilizadas en croma. Para algunas de las secuencias de prueba, cuando cada componente de color se codificó como luma, se observó una reducción de bits de más del 30 % a una calidad comprimida mayor o igual que 45 dB (PSNR promedio).

- 30 Estas y otras características y ventajas de la presente invención pueden ser determinadas fácilmente por un experto en la materia pertinente, en base a las explicaciones de la presente memoria. Se debe entender que las aplicaciones de la presente invención se pueden implementar en varias formas de hardware, software, software inalterable, procesadores de propósito especial o combinaciones de los mismos.

- 35 En el caso más preferente, las explicaciones de la presente invención se implementan como una combinación de hardware y software. Además, el software se puede implementar como un programa de aplicación realizado de forma tangible en una unidad de almacenamiento de programas. El programa de aplicación se puede subir a una máquina que comprende cualquier arquitectura adecuada, y ser ejecutado por la misma. Preferentemente, la máquina se implementa en una plataforma informática que tiene hardware, tal como una o varias unidades centrales de procesamiento (CPU, central processing units), una memoria de acceso aleatorio (RAM, random access memory) e interfaces de entrada/salida (E/S). La plataforma informática puede incluir asimismo un sistema operativo y código de microinstrucciones. Los diversos procesos y funciones descritas en la presente memoria pueden ser parte del código de microinstrucciones o bien parte del programa de aplicación, o cualquier combinación de los mismos, que puede ser ejecutado por una CPU. Además, algunas otras unidades periféricas se pueden conectar a la plataforma informática, tal como una unidad adicional de almacenamiento de datos y una unidad de impresión.
- 45
- 50
- 55

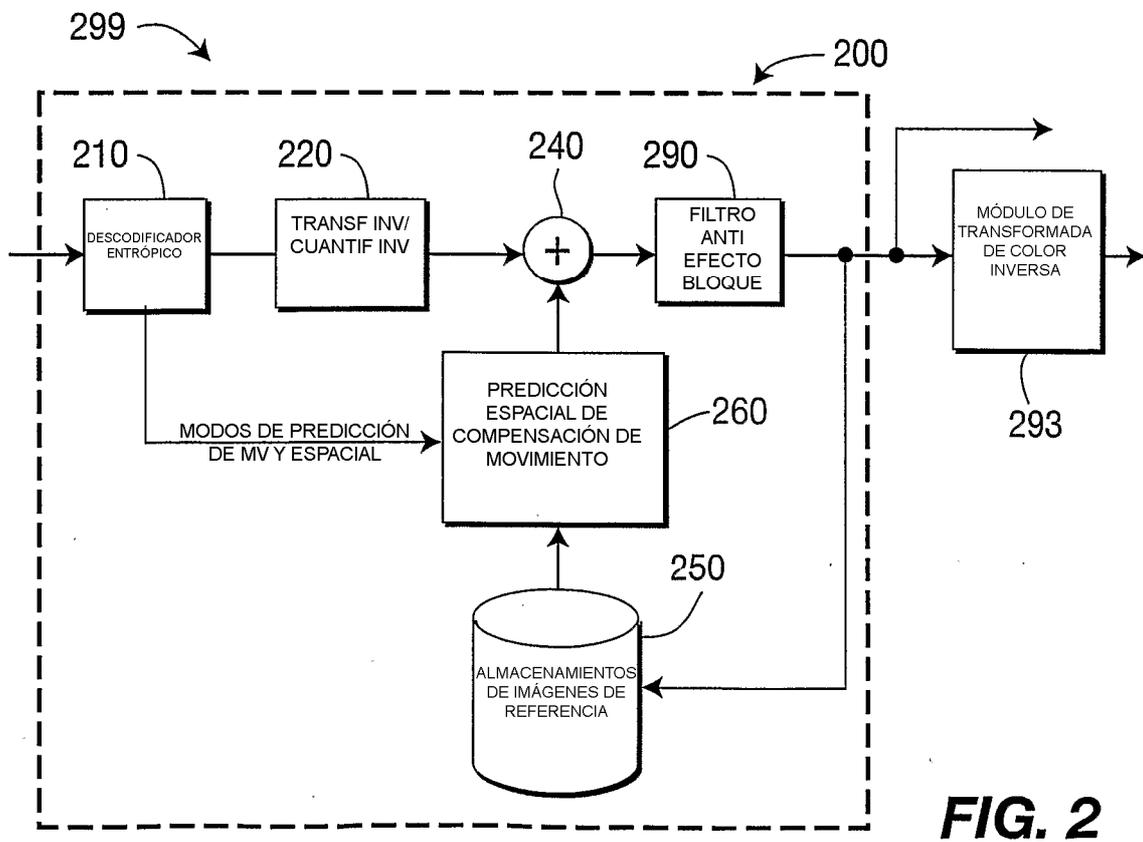
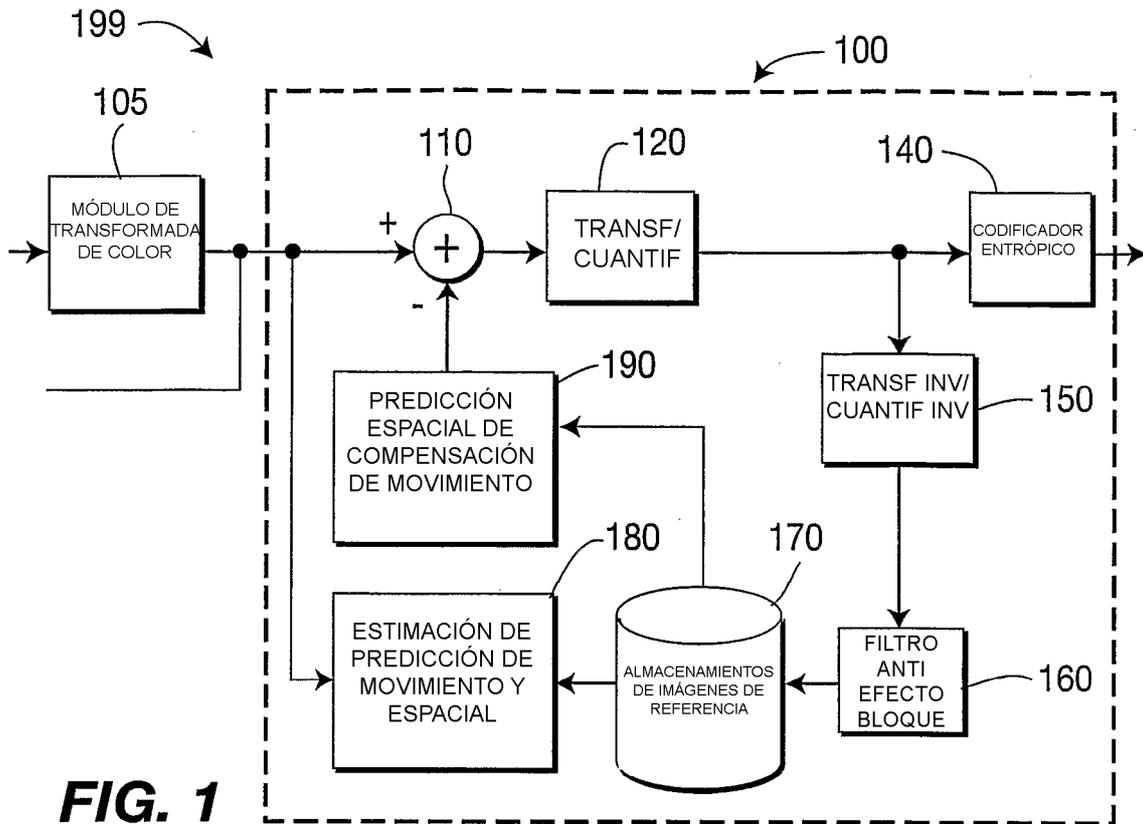
5 Se debe entender además que, debido a que algunos de los procedimientos y componentes de sistema constitutivos representados en los dibujos adjuntos se implementan preferentemente en software, las conexiones reales entre los componentes de sistema o los bloques de función de proceso pueden diferir en función de la manera en la que se programe la presente invención. Dadas las explicaciones de la presente memoria, un experto en la materia pertinente podrá contemplar estas implementaciones o configuraciones de la presente invención y otras similares.

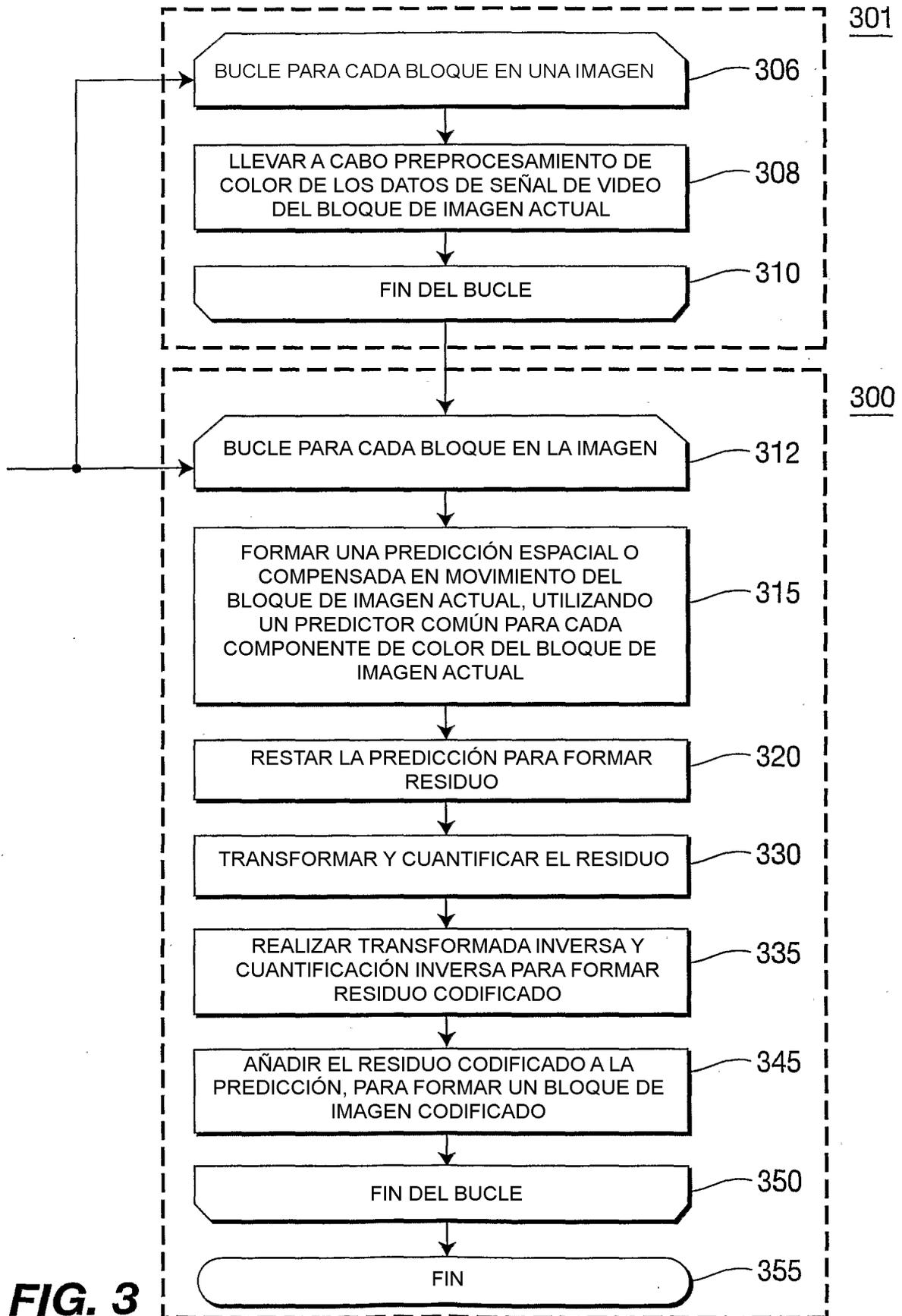
10 Aunque las realizaciones ilustrativas se han descrito en la presente memoria haciendo referencia a los dibujos adjuntos, se debe entender que la presente invención no se limita a dichas realizaciones precisas, y que se pueden efectuar en la misma diversos cambios y modificaciones por un experto en la materia pertinente, sin apartarse del alcance de la presente invención. Se prevé que todos esos cambios y modificaciones se incluyan dentro del alcance de la presente invención tal como se expone en las reivindicaciones adjuntas.

REIVINDICACIONES

1. Un codificador de video para codificar datos de señal de video para un bloque de imagen, comprendiendo el codificador de video un codificador (100) para codificar los datos de señal de video, donde el codificador está configurado para codificar la totalidad de los componentes de color de los datos de señal de video utilizando predictores luma para codificación intra, el codificador está configurado para codificar la totalidad de los componentes de color de los datos de señal de video utilizando un proceso de interpolación luma para los macro bloques B y P para el cálculo de un valor de píxel de referencia en una posición de píxel fraccionaria, y el codificador está configurado para codificar la totalidad de los componentes de color de los datos de señal de video sin aplicar una transformada de color residual a los mismos, y donde el muestreo de los datos de señal de video corresponde a un formato 4:4:4, caracterizado por que los predictores luma para codificación intra se determinan de manera independiente para cada uno de los componentes de color.
2. El codificador de video según la reivindicación 1, en el que un primer modo de predicción espacial para uno primero los componentes de color se ajusta mediante los parámetros `prev_intra4x4_pred_mode_flag0` y `rem_intra4x4_pred_mode0`, un segundo modo de predicción para un segundo de los componentes de color se ajusta mediante los parámetros `prev_intra4x4_pred_mode_flag1` y `rem_intra4x4_pred_mode1` y un tercer modo de predicción para un tercero de los componentes de color se ajusta mediante los parámetros `prev_intra4x4_pred_mode_flag2` y `rem_intra4x4_pred_mode2` del estándar H.264 del sector de telecomunicaciones de la Unión Internacional de Telecomunicaciones.
3. Un procedimiento para codificar datos de señal de video para un bloque de imagen, comprendiendo el procedimiento codificar (1415) los datos de señal de video, donde dicha codificación codifica la totalidad de los componentes de color de los datos de señal de video utilizando predictores luma para codificación intra, dicha codificación codifica la totalidad de los componentes de color de los datos de señal de video utilizando un proceso de interpolación luma para los macro bloques B y P para el cálculo de un valor de píxel de referencia en una posición de píxel fraccionaria, y dicha codificación codifica la totalidad de los componentes de color de los datos de señal de video sin aplicar una transformada de color residual a los mismos y el muestreo de los datos de señal de video corresponde a un formato 4:4:4, caracterizado por que los predictores luma para codificación intra se determinan independientemente para cada uno de los componentes de color.
4. El procedimiento según la reivindicación 3, en el que un primer modo de predicción espacial para uno primero de los componentes de color se ajusta mediante los parámetros `prev_intra4x4_pred_mode_flag0` y `rem_intra4x4_pred_mode0`, un segundo modo de predicción para un segundo de los componentes de color se ajusta mediante los parámetros `prev_intra4x4_pred_mode_flag1` y `rem_intra4x4_pred_mode1` y un tercer modo de predicción para un tercero de los componentes de color se ajusta mediante los parámetros `prev_intra4x4_pred_mode_flag2` y `rem_intra4x4_pred_mode2` del estándar H.264 del sector de telecomunicaciones de la Unión Internacional de Telecomunicaciones.
5. Un descodificador de video para descodificar datos de señal de video para un bloque de imagen, comprendiendo el descodificador de video un descodificador (200) para descodificar los datos de señal de video, donde dicho descodificador está configurado para descodificar la totalidad de los componentes de color de los datos de señal de video utilizando predictores luma para codificación intra, dicho descodificador está configurado para descodificar la totalidad de los componentes de color de los datos de señal de video utilizando un proceso de interpolación luma para los macrobloques B y P para el cálculo de un valor de píxel de referencia en una posición de píxel fraccionaria, y dicho descodificador está configurado para descodificar la totalidad de los componentes de color de los datos de señal de video sin aplicar una transformada de color residual a los mismos y muestrear los datos de señal de video corresponde a un formato 4:4:4, caracterizado por que los predictores luma para codificación intra se determinan independientemente para cada uno de los componentes de color.
6. El descodificador de video según la reivindicación 5, en el que un primer modo de predicción espacial para uno primero los componentes de color se ajusta mediante los parámetros `prev_intra4x4_pred_mode_flag0` y `rem_intra4x4_pred_mode0`, un segundo modo de predicción para un segundo de los componentes de color se ajusta mediante los parámetros `prev_intra4x4_pred_mode_flag1` y `rem_intra4x4_pred_mode1` y un tercer modo de predicción para un tercero de los componentes de color se ajusta mediante los parámetros `prev_intra4x4_pred_mode_flag2` y `rem_intra4x4_pred_mode2` del estándar H.264 del sector de telecomunicaciones de la Unión Internacional de Telecomunicaciones.
7. Un procedimiento para descodificar datos de señal de video para un bloque de imagen, comprendiendo el procedimiento descodificar (1415) los datos de señal de video, donde dicha descodificación descodifica la totalidad de los componentes de color de los datos de señal de video utilizando predictores luma para codificación intra, dicha descodificación descodifica la totalidad de los componentes de color de los datos de señal de video utilizando un proceso de interpolación luma para los macro bloques B y P para el cálculo de un valor de píxel de referencia en una posición de píxel fraccionaria, y dicha descodificación descodifica la totalidad de los componentes de color de los datos de señal de video sin aplicar una transformada de color residual a los mismos y el muestreo de los datos de señal de video corresponde a un formato 4:4:4, caracterizado por que los predictores luma para codificación intra se determinan independientemente para cada uno de los componentes de color.

- 5 8. El procedimiento según la reivindicación 7, en el que un primer modo de predicción espacial para un primero de los componentes de color se ajusta mediante los parámetros `prev_intra4x4_pred_mode_flag0` y `rem_intra4x4_pred_mode0`, un segundo modo de predicción para un segundo de los componentes de color se ajusta mediante los parámetros `prev_intra4x4_pred_mode_flag1` y `rem_intra4x4_pred_mode1` y un tercer modo de predicción para un tercero de los componentes de color se ajusta mediante los parámetros `prev_intra4x4_pred_mode_flag2` y `rem_intra4x4_pred_mode2` del estándar H.264 del sector de telecomunicaciones de la Unión Internacional de Telecomunicaciones.





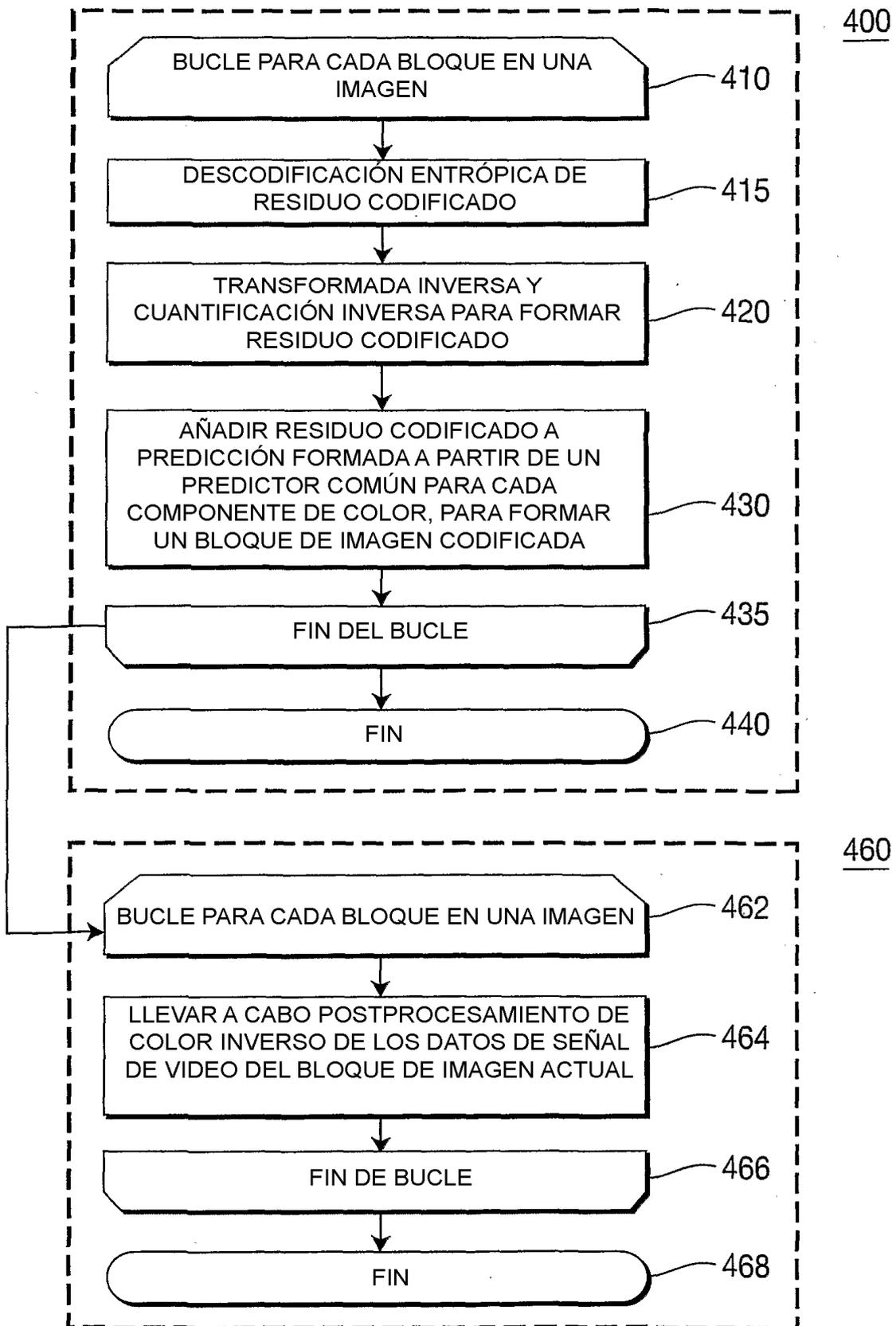


FIG. 4

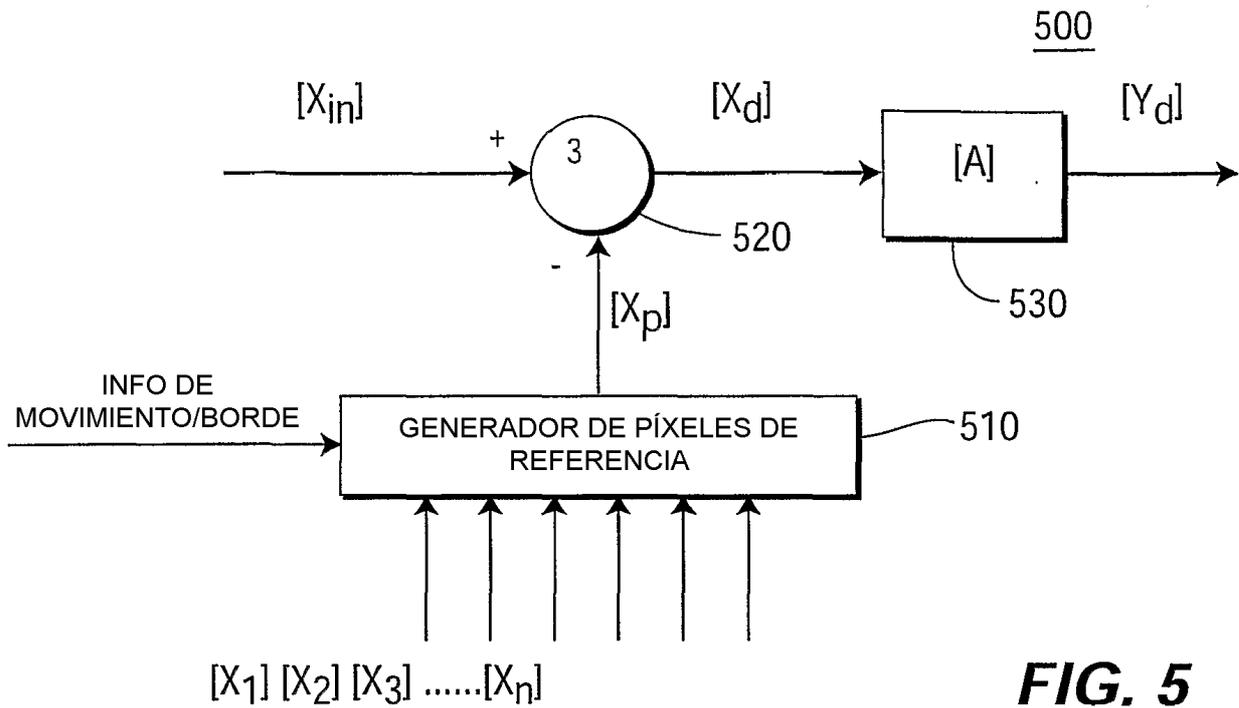


FIG. 5

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)
Si (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

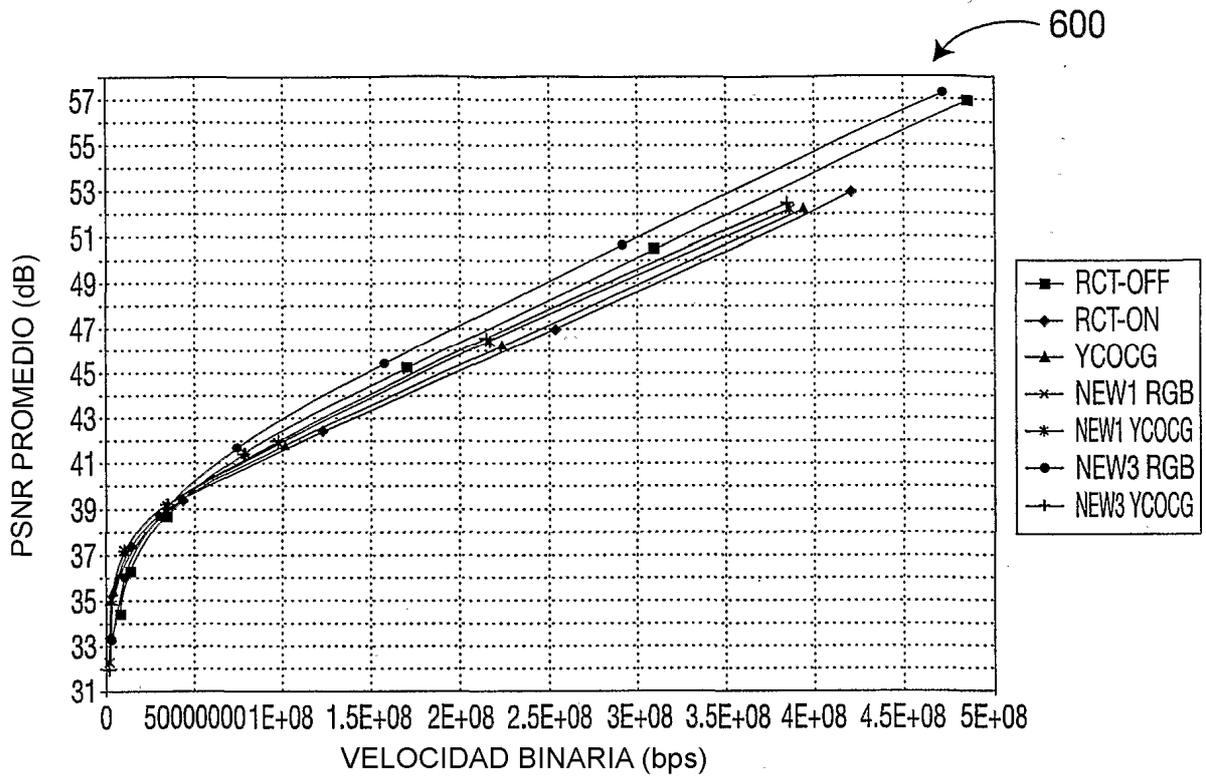


FIG. 6A

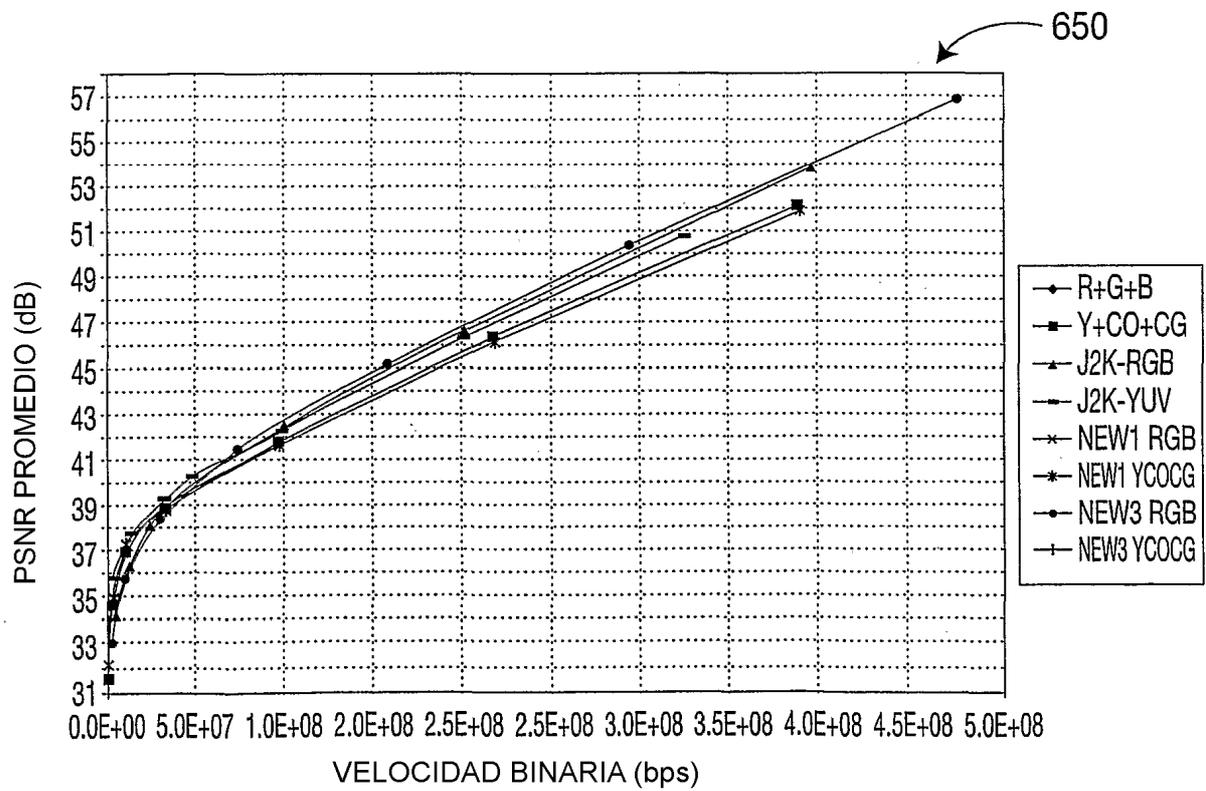


FIG. 6B

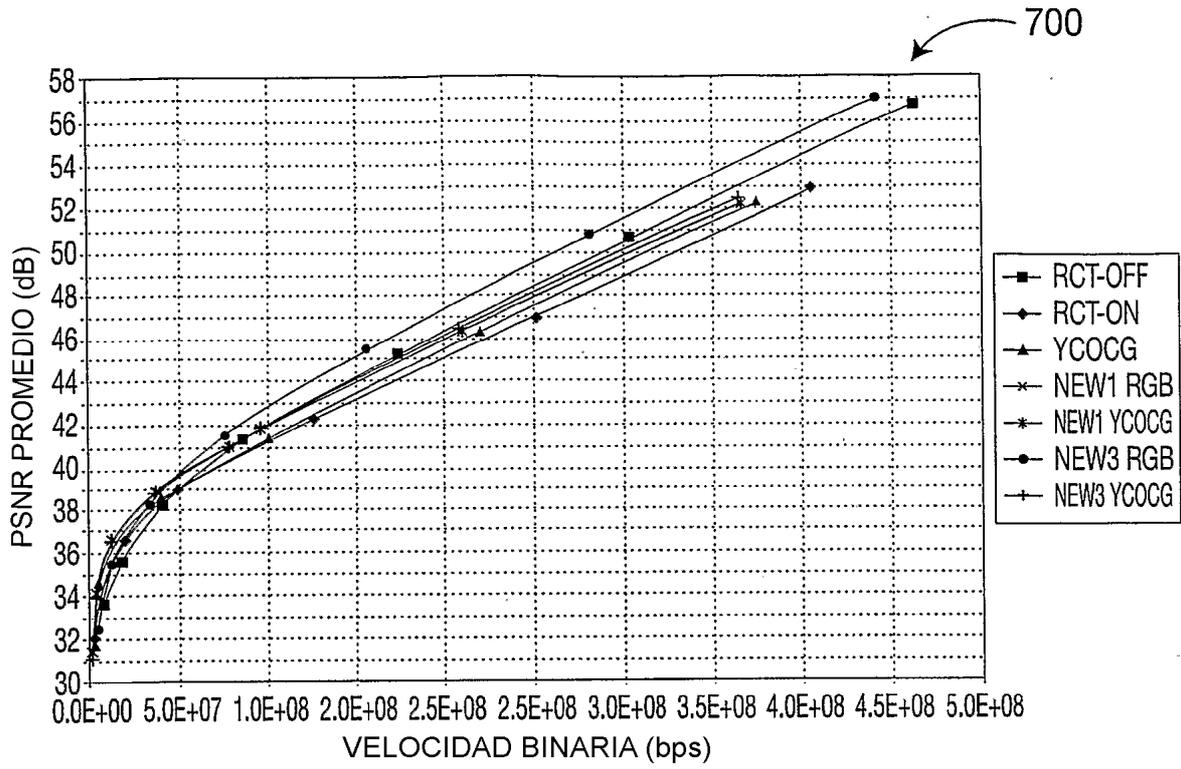


FIG. 7A

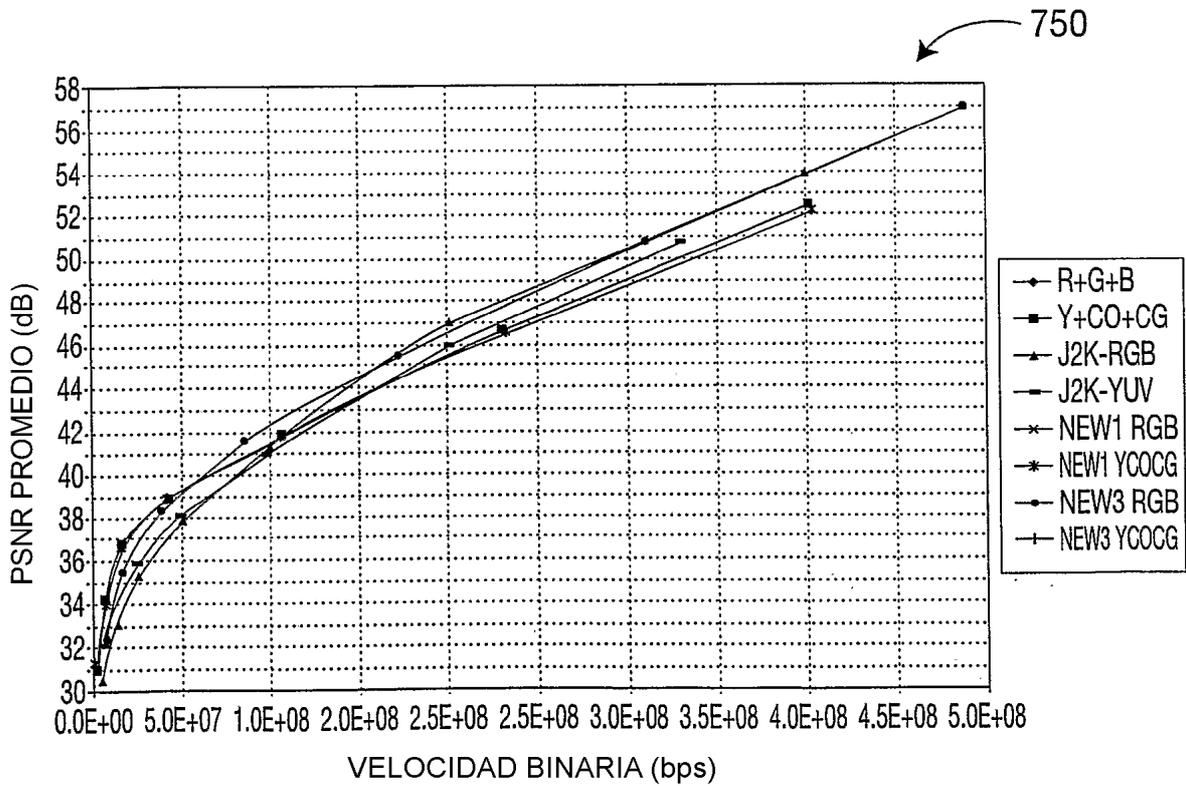


FIG. 7B

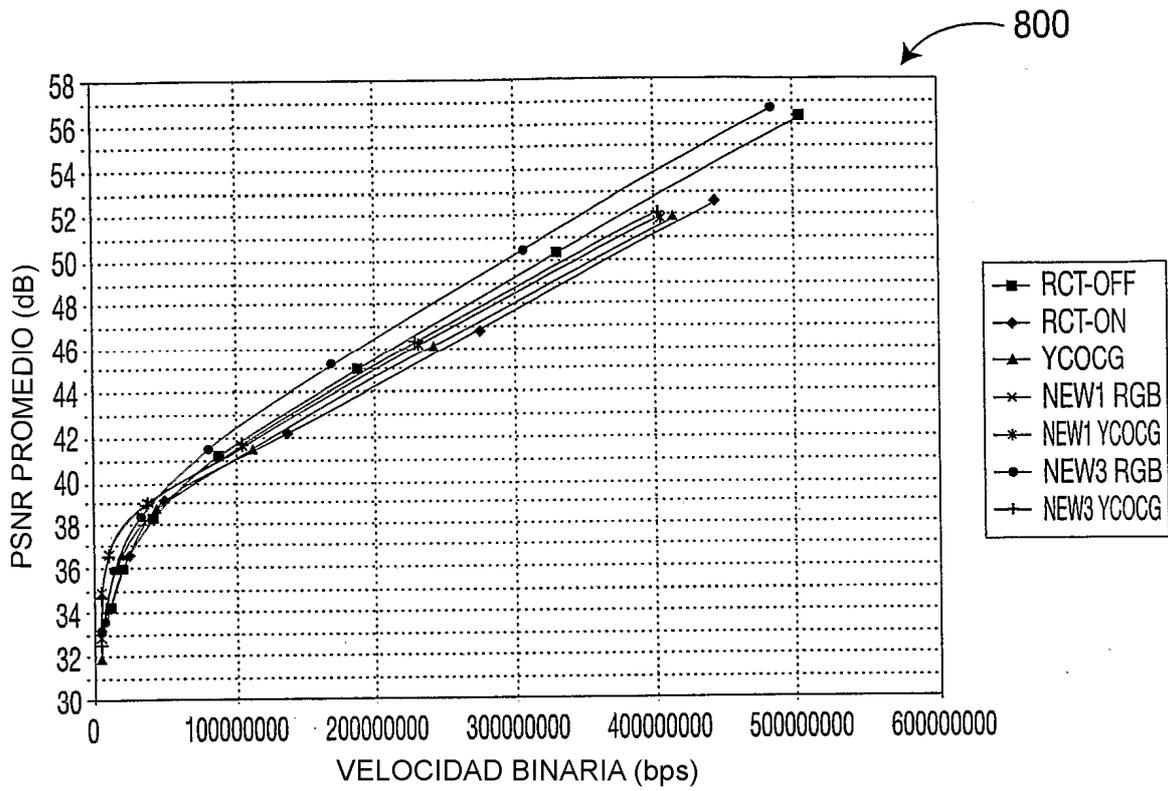


FIG. 8A

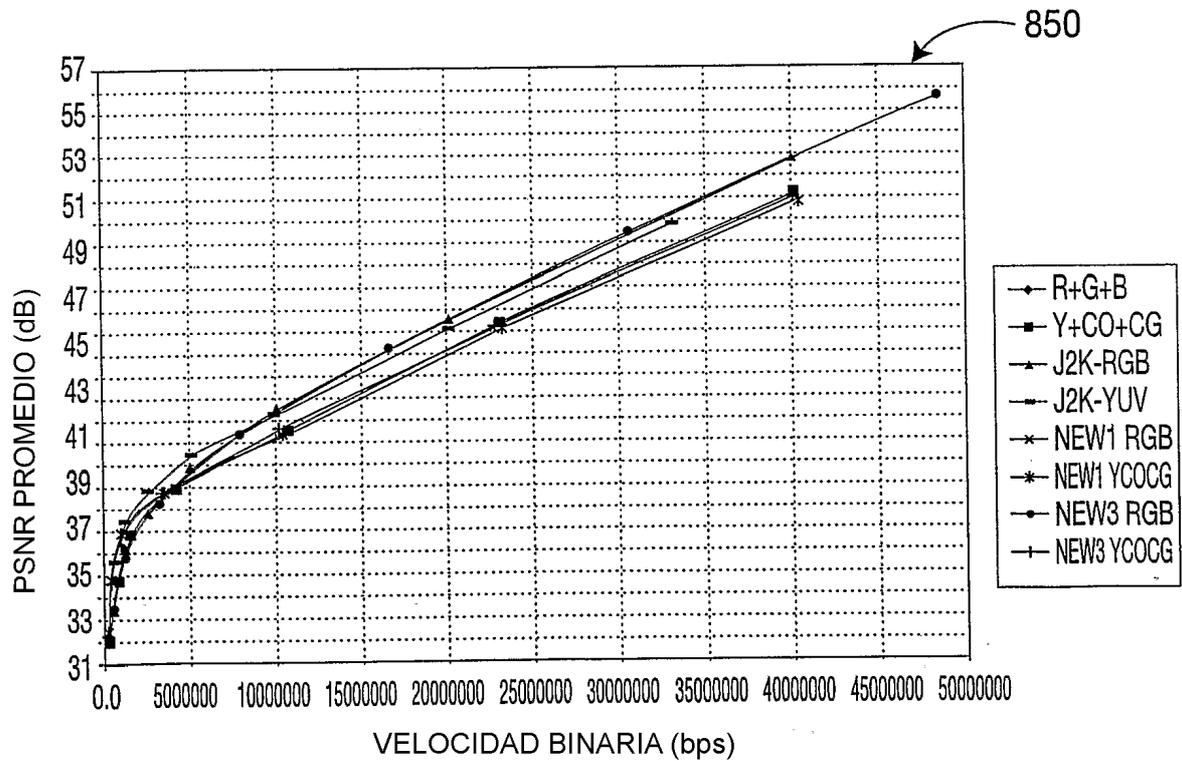


FIG. 8B

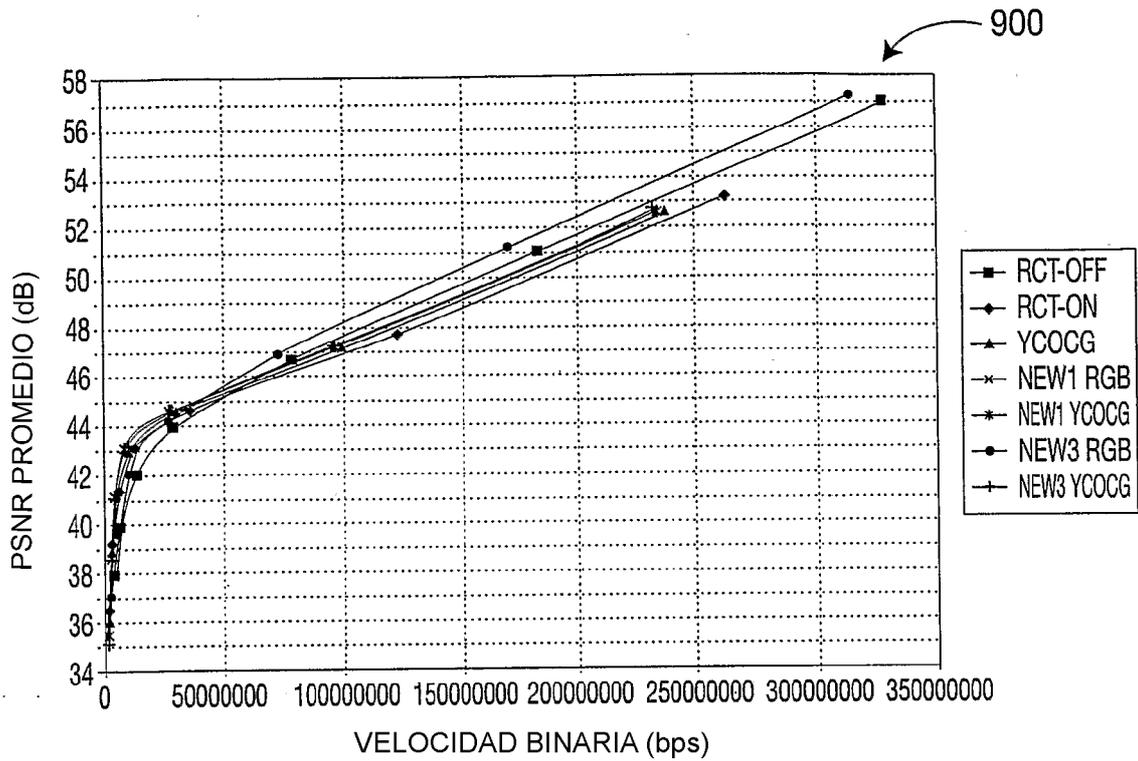


FIG. 9A

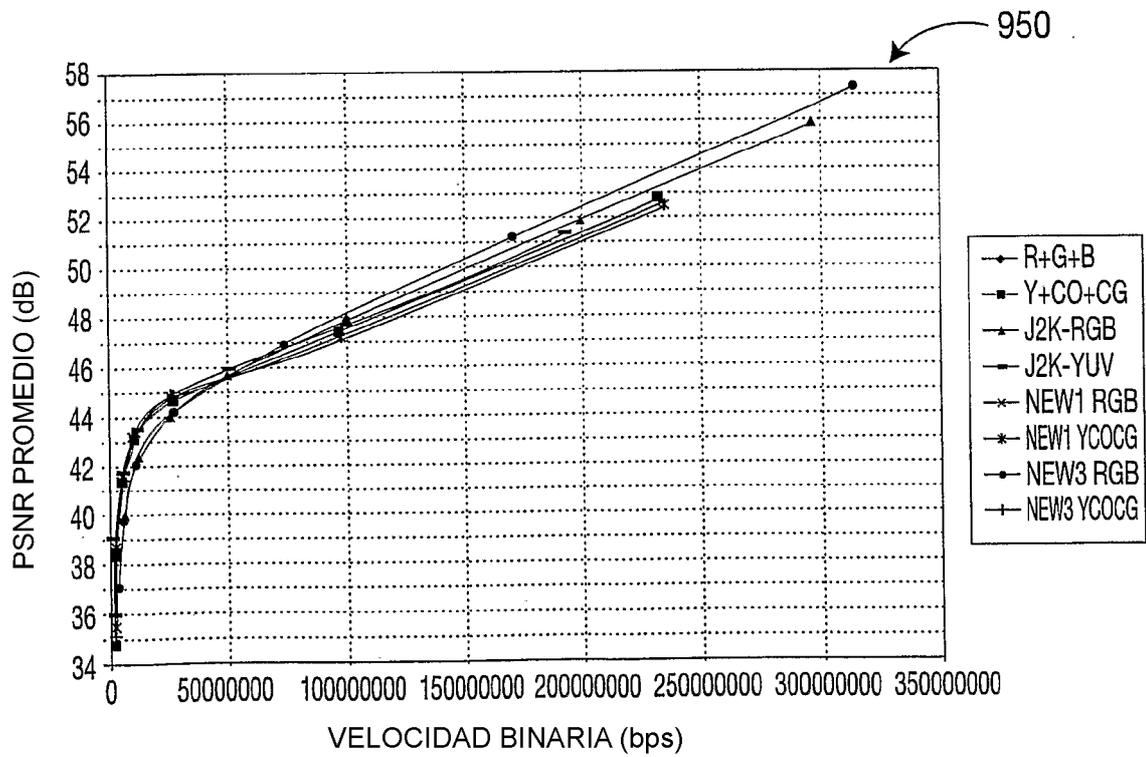


FIG. 9B

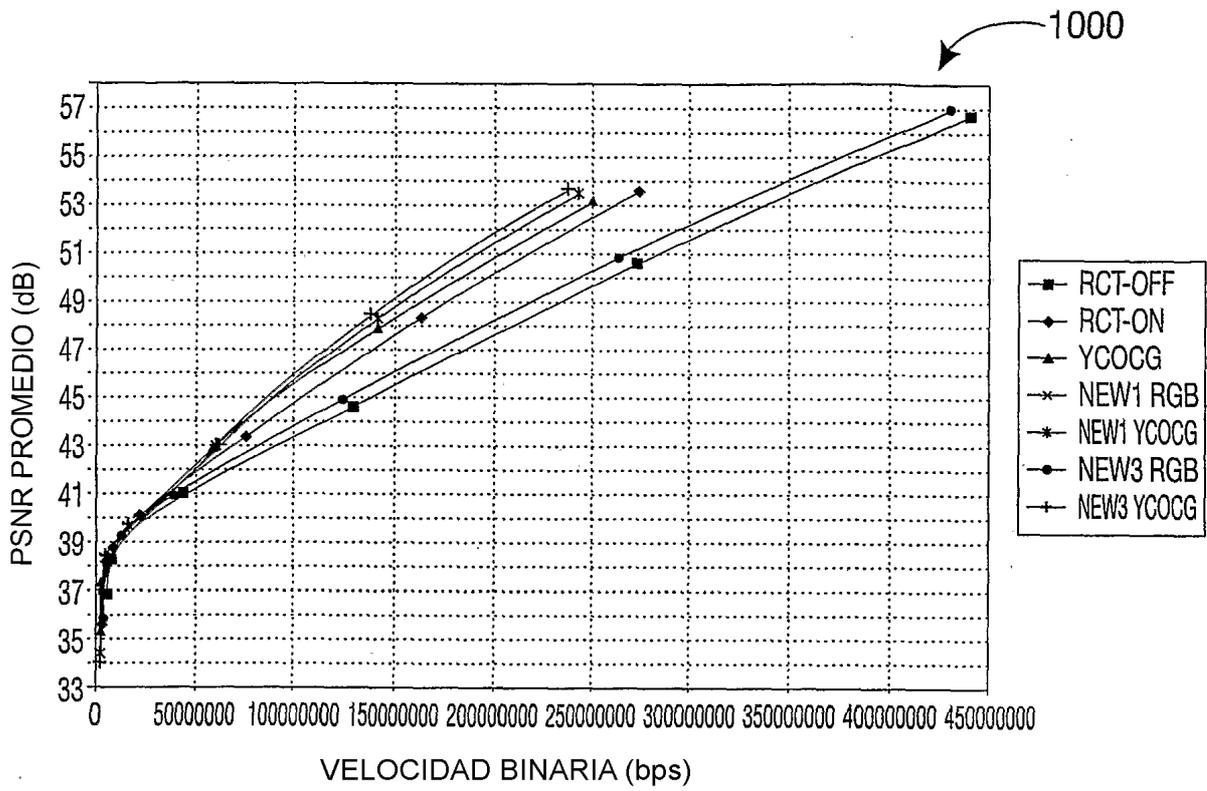


FIG. 10A

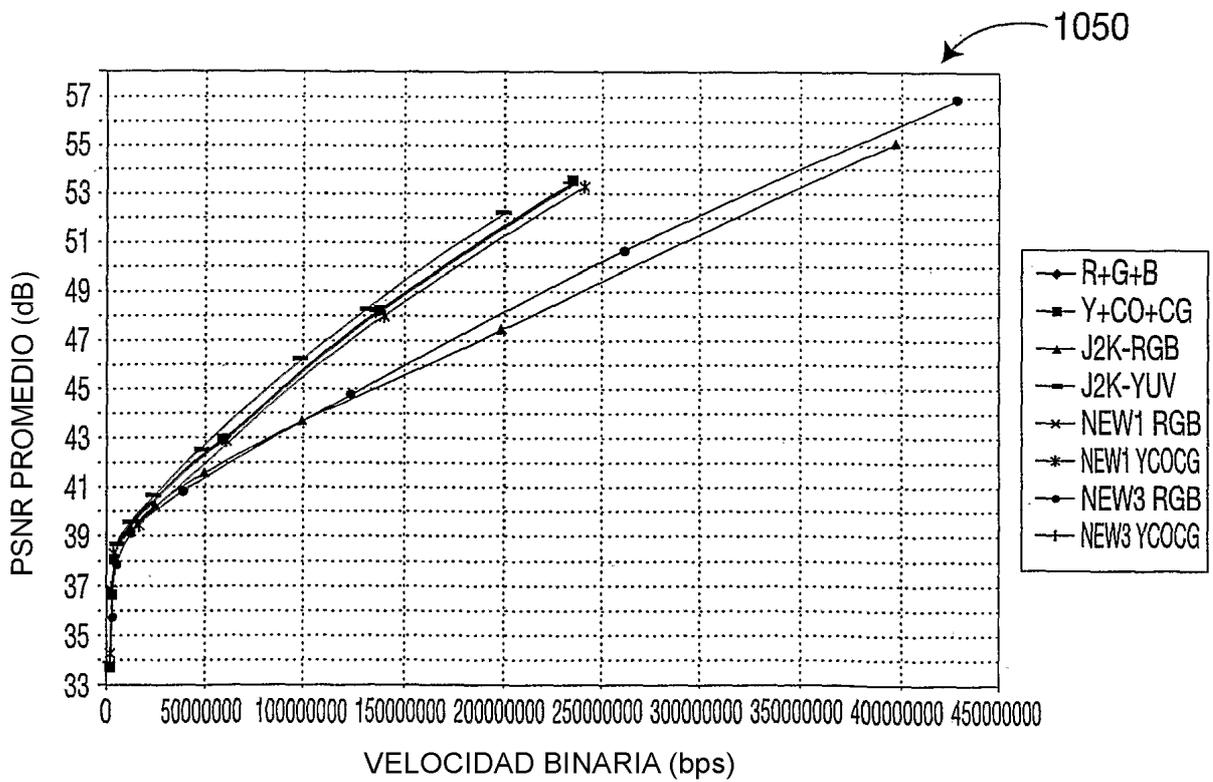


FIG. 10B

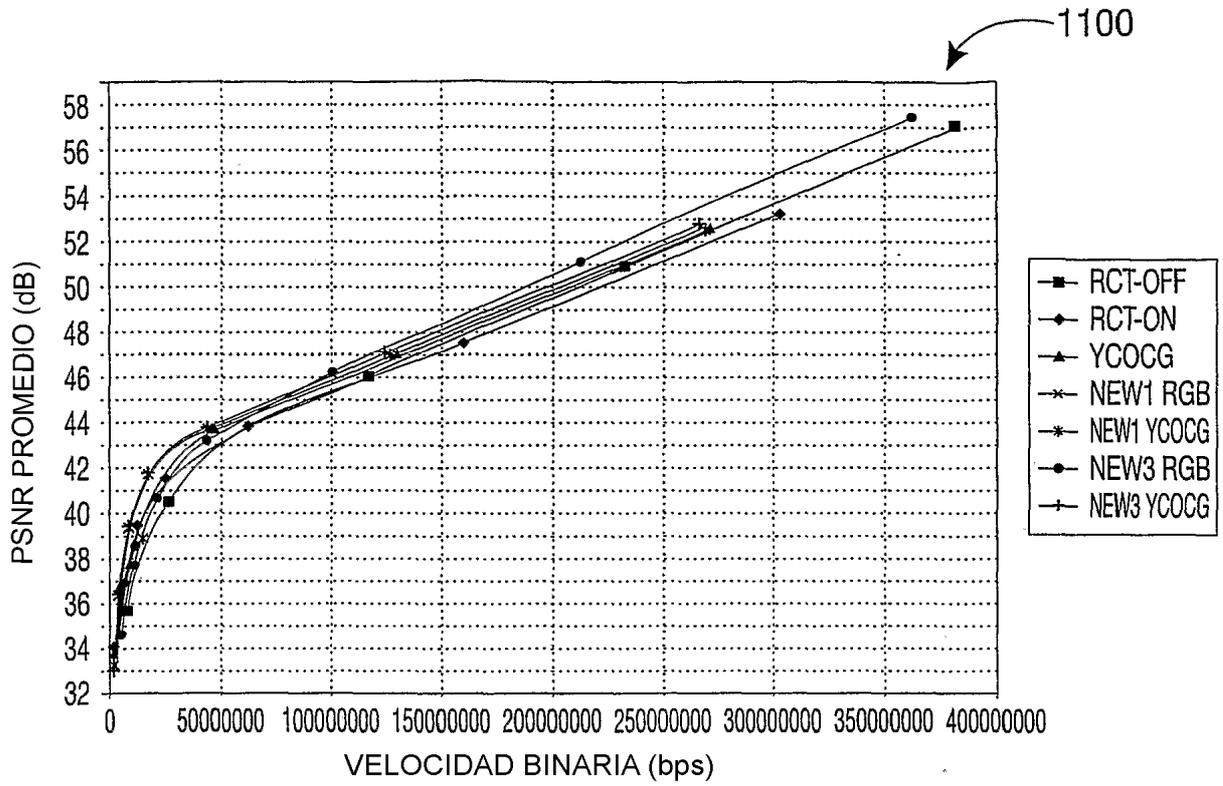


FIG. 11A

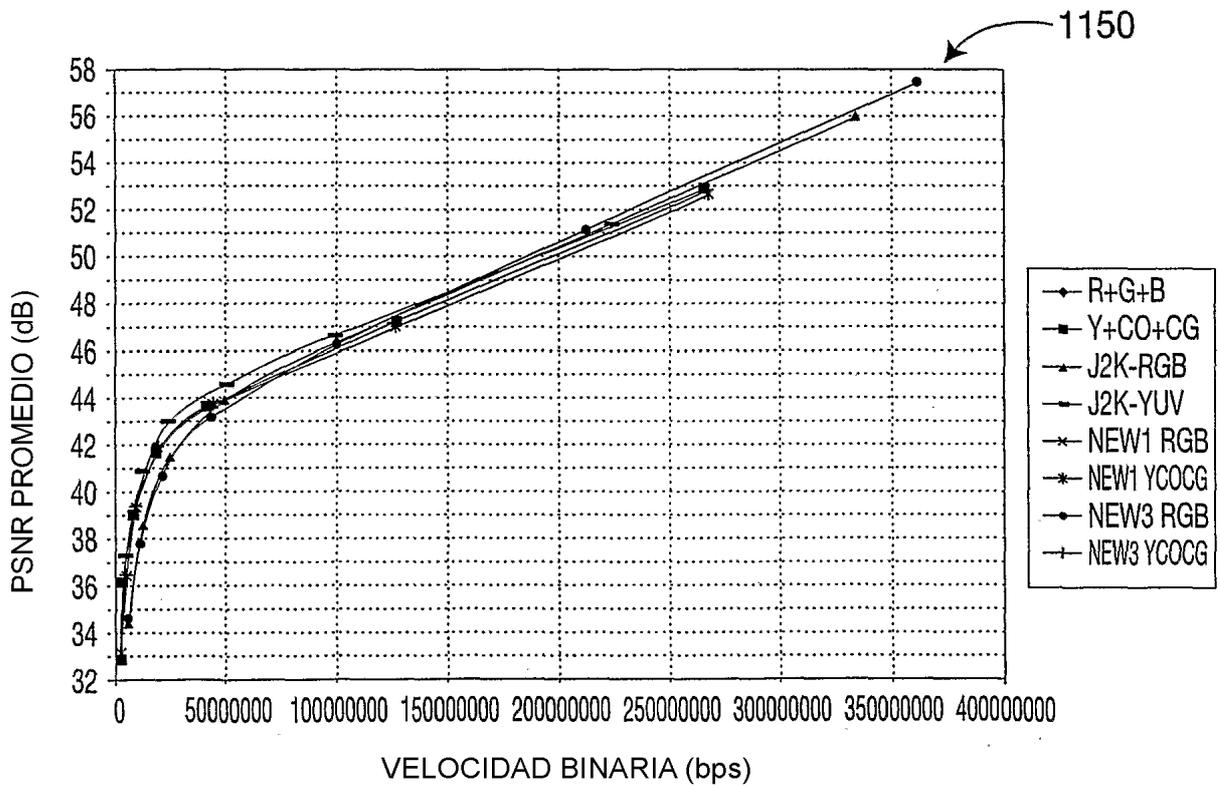


FIG. 11B

FIG. 13

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

FIG. 13A

Residual() {	C	Descriptor
Si(!entropy_coding_mode_flag)		
residual_block = residual_block_cavlc		
si no		
residual_block = residual_block_cabac		
Si(MbPartPredMode(mb_type, 0) == Intra_16x16)		
residual_block(Intra16x16DCLevel, 16)	3	
para(i8x8 = 0; i8x8 < 4; i8x8++) /* each luma 8x8 block */		
Si(!transform_size_8x8_flag !entropy_coding_mode_flag)		
para(i4x4 = 0; i4x4 < 4; i4x4++) { /* each 4x4 sub-block of block */		
Si(CodedBlockPatternLuma & (1 << i8x8))		
Si(MbPartPredMode(mb_type, 0) == Intra_16x16)		
residual_block(Intra16x16ACLevel[i8x8 * 4 + i4x4], 15)	3	
si no		
residual_block(LumaLevel[i8x8 * 4 + i4x4], 16)	3 4	
si no, si (MbPartPredMode(mb_type, 0) == Intra_16x16)		
para(i = 0; i < 15; i++)		
Intra16x16ACLevel[i8x8 * 4 + i4x4][i] = 0		
si no		
para(i = 0; i < 16; i++)		
LumaLevel[i8x8 * 4 + i4x4][i] = 0		
Si(!entropy_coding_mode_flag && transform_size_8x8_flag)		
para(i = 0; i < 16; i++)		
LumaLevel8x8[i8x8][4 * i + i4x4] = LumaLevel[i8x8 * 4 + i4x4][i]		
}		

FIG. 13B

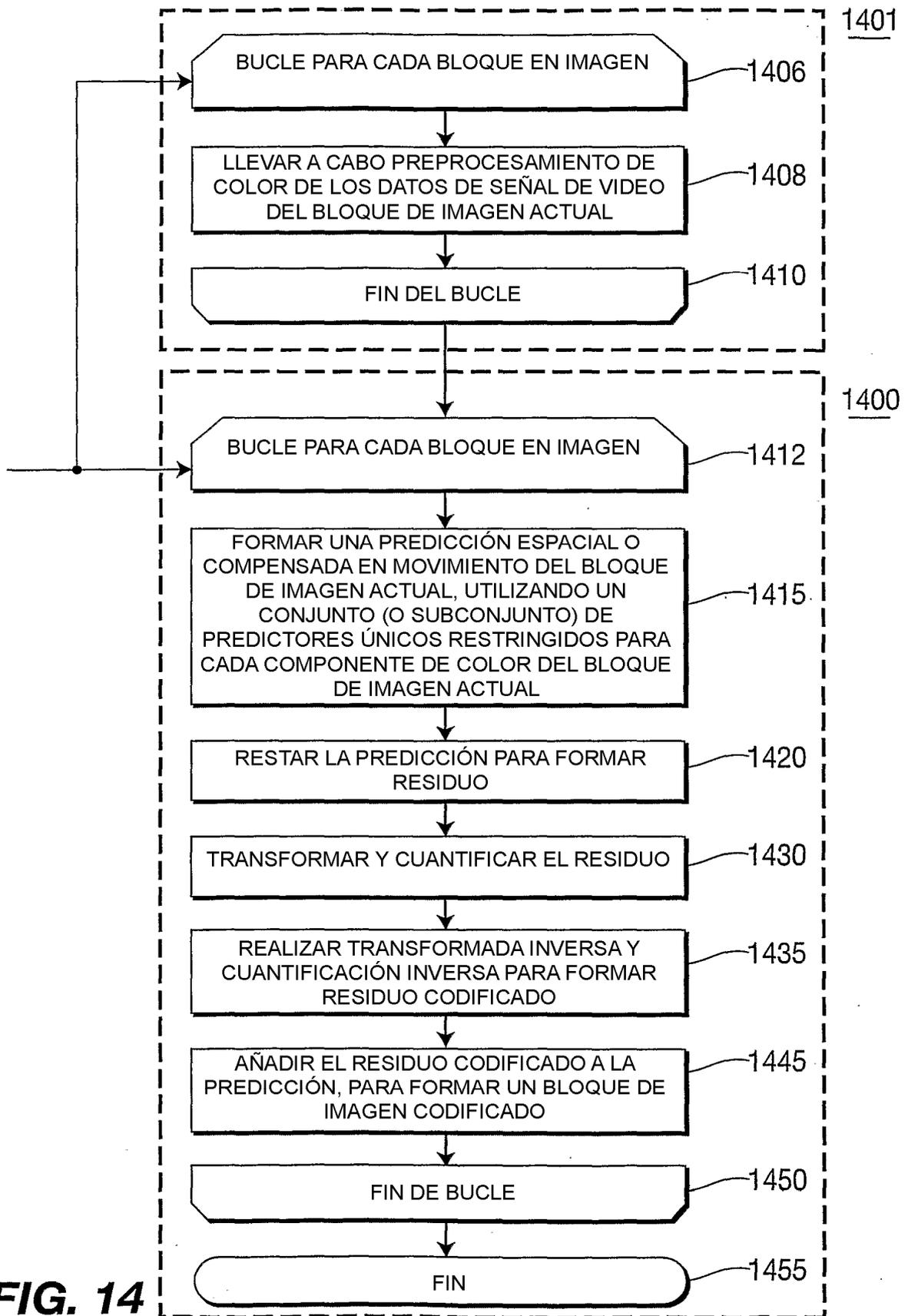
si no, si(CodedBlockPatternLuma & (1 << i8x8)		
residual_block(LumaLevel8x8[i8x8], 64)	3 4	
si no		
para (i = 0; i < 64; i++)		
LumaLevel8x8[i8x8][i] = 0		
si(chroma_format_idc == 1 chroma_format_idc == 2) {		
NumC8x8 = 4 / (SubWidthC * SubHeightC)		
para (iCbCr = 0; iCbCr < 2; iCbCr++)		
si (CodedBlockPatternChroma & 3) /* chroma DC residual present*/		
residual_block(ChromaDCLevel[iCbCr], 4 * NumC8x8)	3 4	
si no		
para (i = 0; i < 4 * NumC8x8; i++)		
ChromaDCLevel[iCbCr][i] = 0		
para (iCbCr = 0; iCbCr < 2; iCbCr++)		
para (i8x8 = 0; i8x8 < NumC8x8; i8x8++)		
para(i4x4 = 0; i4x4 < 4; i4x4++)		
si (CodedBlockPatternChroma & 2) /* chroma AC residual present */		
residual_block(ChromaACLevel[iCbCr][i8x8*4+i4x4], 15)	3 4	
si no		
para (i = 0; i < 15; i++)		
ChromaACLevel[iCbCr][i8x8*4+i4x4][i] = 0		
}si no, si(chroma_format_idc == 3) {		
si(MbPartPredMode(mb_type, 0) == Intra_16x16) /*Cb for 4:4:4*/		
residual_block(CbIntra16x16DCLevel, 16)	3	
para(i8x8 = 0; i8x8 < 4; i8x8++) /* each Cb 8x8 block */		
si (!transform_size_8x8_flag !entropy_coding_mode_flag)		
para (i4x4 = 0; i4x4 < 4; i4x4++) { /* each 4x4 sub-block of block */		
si (CodedBlockPatternLuma & (1 << i8x8)		

FIG. 13C

Si (MbPartPredMode(mb_type, 0) == Intra_16x16)		
residual_block(CbIntra16x16ACLevel[i8x8*4 + i4x4], 15)	3	
si no		
si no, si residual_block(CbLevel[i8x8 * 4 + i4x4], 16)	3 4	
para (MbPartPredMode(mb_type, 0) == Intra_16x16)		
(i = 0; i < 15; i++)		
CbIntra16x16ACLevel[i8x8 * 4 + i4x4][i] = 0		
si no		
para (i = 0; i < 16; i++)		
CbLevel[i8x8 * 4 + i4x4][i] = 0		
Si (!entropy_coding_mode_flag && transform_size_8x8_flag)		
para (i = 0; i < 16; i++)		
CbLevel8x8[i8x8][4 * i + i4x4] = CbLevel[i8x8 * 4 + i4x4][i]		
}		
si no, si (CodedBlockPatternLuma & (1 << i8x8))		
residual_block(CbLevel8x8[i8x8], 64)	3 4	
si no		
para (i = 0; i < 64; i++)		
CbLevel8x8[i8x8][i] = 0		
Si (MbPartPredMode(mb_type, 0) == Intra_16x16) /*Cr for 4:4:4 */		
residual_block(CrIntra16x16DCLevel, 16)	3	
para (i8x8 = 0; i8x8 < 4; i8x8++) /* each Cr 8x8 block */		
Si (!transform_size_8x8_flag !entropy_coding_mode_flag)		
para (i4x4 = 0; i4x4 < 4; i4x4++) { /* each 4x4 sub-block of block */		
Si (CodedBlockPatternLuma & (1 << i8x8))		
Si (MbPartPredMode(mb_type, 0) == Intra_16x16)		
residual_block(CrIntra16x16ACLevel[i8x8*4+i4x4], 15)	3	
si no		

FIG. 13D

residual_block(CrLevel[i8x8 * 4 + i4x4], 16)	3 4	
si no, si (MbPartPredMode(mb_type, 0) == Intra_16x16)		
para (i = 0; i < 15; i++)		
CrIntra16x16ACLevel[i8x8 * 4 + i4x4][i] = 0		
si no		
para (i = 0; i < 16; i++)		
CrLevel[i8x8 * 4 + i4x4][i] = 0		
si (!entropy_coding_mode_flag && transform_size_8x8_flag)		
para (i = 0; i < 16; i++)		
CrLevel8x8[i8x8][4 * i + i4x4] = CrLevel[i8x8 * 4 + i4x4][i]		
}		
si no, si (CodedBlockPatternLuma & (1 << i8x8))		
residual_block(CrLevel8x8[i8x8], 64)	3 4	
si no		
para (i = 0; i < 64; i++)		
CrLevel8x8[i8x8][i] = 0		
}		



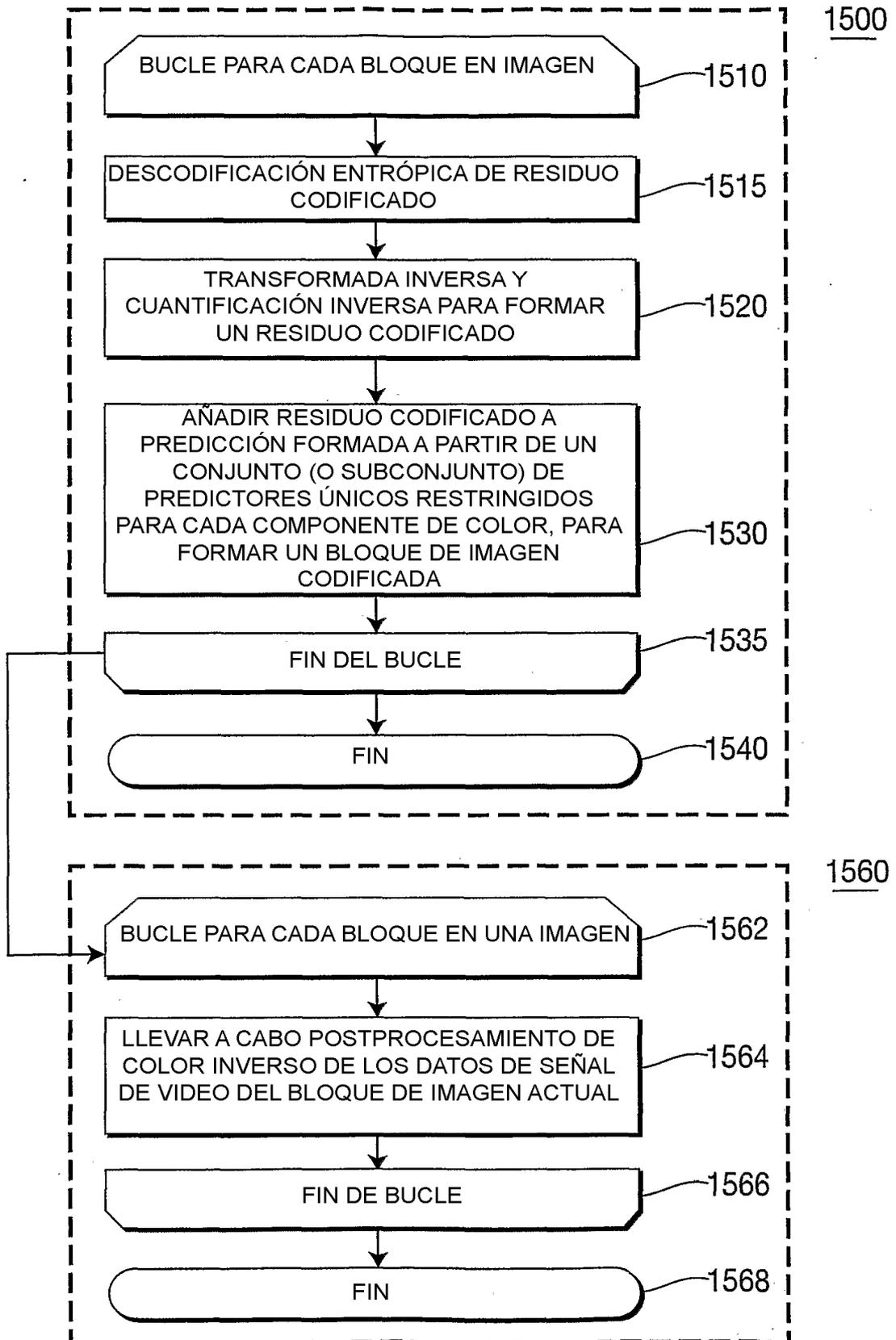


FIG. 15

FIG. 16

FIG. 16A

FIG. 16B

1600

FIG. 16A

mb_pred(mb_type) {	C	Descriptor
Si (MbPartPredMode(mb_type, 0) == Intra_4x4 MbPartPredMode(mb_type, 0) == Intra_8x8 MbPartPredMode(mb_type, 0) == Intra_16x16) {		
Si (profile_idc != 166) {		
Si (MbPartPredMode(mb_type, 0) == Intra_4x4)		
para (luma4x4BlkIdx=0; luma4x4BlkIdx<16; luma4x4BlkIdx++) {		
prev_intra4x4_pred_mode_flag[luma4x4BlkIdx]	2	u(1) ae(v)
Si (!prev_intra4x4_pred_mode_flag[luma4x4BlkIdx])		
rem_intra4x4_pred_mode[luma4x4BlkIdx]	2	u(3) ae(v)
}		
Si (MbPartPredMode(mb_type, 0) == Intra_8x8)		
para (luma8x8BlkIdx=0; luma8x8BlkIdx<4; luma8x8BlkIdx++) {		
prev_intra8x8_pred_mode_flag[luma8x8BlkIdx]	2	u(1) ae(v)
Si (!prev_intra8x8_pred_mode_flag[luma8x8BlkIdx])		
rem_intra8x8_pred_mode[luma8x8BlkIdx]	2	u(3) ae(v)
}		
Si (chroma_format_idc != 0)		
intra_chroma_pred_mode	2	u(v) ae(v)
} si no {		
Si (MbPartPredMode(mb_type, 0) == Intra_4x4)		
para (luma4x4BlkIdx=0; luma4x4BlkIdx<16; luma4x4BlkIdx++) {		
prev_intra4x4_pred_mode_flag0[luma4x4BlkIdx]	2	u(1) ae(v)
Si (!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

si (!prev_intra4x4_pred_mode_flag1[luma4x4BlkIdx])		
rem_intra4x4_pred_mode1[luma4x4BlkIdx]	2	u(3) ae(v)
prev_intra4x4_pred_mode_flag2[luma4x4BlkIdx]	2	u(1) ae(v)
si(!prev_intra4x4_pred_mode_flag2[luma4x4BlkIdx])		
rem_intra4x4_pred_mode2[luma4x4BlkIdx]	2	u(3) ae(v)
}		
si(MbPartPredMode(mb_type, 0) == Intra_8x8)		
para (luma8x8BlkIdx=0; luma8x8BlkIdx<4; luma8x8BlkIdx++) {		
prev_intra8x8_pred_mode_flag0[luma8x8BlkIdx]	2	u(1) ae(v)
si(!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)
si(!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)
si(!prev_intra8x8_pred_mode_flag2[luma8x8BlkIdx])		
rem_intra8x8_pred_mode2[luma8x8BlkIdx]	2	u(3) ae(v)
}		
}		
}si no, si(MbPartPredMode(mb_type, 0) != Direct) {		
.....		