

(19)



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS  
ESPAÑA



(11) Número de publicación: **2 860 448**

(51) Int. Cl.:

**H04N 19/51** (2014.01)  
**H04N 19/583** (2014.01)  
**H04N 19/70** (2014.01)  
**H04N 19/134** (2014.01)  
**H04N 19/109** (2014.01)  
**H04N 19/503** (2014.01)

(12)

TRADUCCIÓN DE PATENTE EUROPEA

T3

(86) Fecha de presentación y número de la solicitud internacional: **23.09.2013 PCT/US2013/061241**

(87) Fecha y número de publicación internacional: **03.04.2014 WO14052249**

(96) Fecha de presentación y número de la solicitud europea: **23.09.2013 E 13774010 (6)**

(97) Fecha y número de publicación de la concesión europea: **03.03.2021 EP 2901699**

---

(54) Título: **Señalización condicional de información de modificación de lista de imágenes de referencia**

(30) Prioridad:

**30.09.2012 US 201261708042 P  
28.02.2013 US 201313781710**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:  
**05.10.2021**

(73) Titular/es:

**MICROSOFT TECHNOLOGY LICENSING, LLC  
(100.0%)  
One Microsoft Way  
Redmond, WA 98052 , US**

(72) Inventor/es:

**SULLIVAN, GARY J. y  
KANUMURI, SANDEEP**

(74) Agente/Representante:

**ELZABURU, S.L.P**

**ES 2 860 448 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

Señalización condicional de información de modificación de lista de imágenes de referencia

### Antecedentes

Los ingenieros utilizan la compresión (también llamada codificación de fuente o codificación de fuente) para reducir la tasa de bits del video digital. La compresión reduce el coste de almacenar y transmitir información de video al convertir la información en una forma de tasa de bits más baja. La descompresión (también llamada decodificación) reconstruye una versión de la información original a partir de la forma comprimida. Un "códec" es un sistema codificador/decodificador.

Durante las últimas dos décadas, se han adoptado varios estándares de códecs de video, incluidos los estándares H.261, H.262 (MPEG-2 o ISO/IEC 13818-2), H.263 y H.264 (AVC o ISO/IEC 14496-10) y los estándares MPEG-1 (ISO/IEC 11172-2), MPEG-4 Visual (ISO/IEC 14496-2) y SMPTE 421M. Más recientemente, el estándar HEVC está en desarrollo y una versión reciente es: Bross et al., "High Efficiency Video Coding (HEVC) text specification draft 8", 10. REUNIÓN JCT-VC; 101. REUNIÓN MPEG; 11-7-2012 – 20-7-2012; Estocolmo. Un estándar de códec de video generalmente define opciones para la sintaxis de un flujo de bits de video codificado, al detallar los parámetros en el flujo de bits cuando se utilizan características particulares en la codificación y decodificación. En muchos casos, un códec de video estándar también proporciona detalles sobre las operaciones de decodificación que debe realizar un decodificador para lograr resultados correctos en la decodificación. Aparte de los estándares de códec, varios formatos de códec patentados definen otras opciones para la sintaxis de un flujo de bits de video codificado y las operaciones de decodificación correspondientes.

Algunos tipos de parámetros en un flujo de bits indican información sobre imágenes de referencia utilizadas durante la codificación y decodificación de vídeo. Una imagen de referencia es, en general, una imagen que contiene muestras que pueden usarse para la predicción entre imágenes en el proceso de decodificación de otras imágenes. Normalmente, las otras imágenes siguen la imagen de referencia en el orden de decodificación y utilizan la imagen de referencia para la predicción con compensación de movimiento. En algunos formatos y estándares de códec de vídeo, se encuentran disponibles múltiples imágenes de referencia en un momento dado para su uso en la predicción con compensación de movimiento. Dichos formatos/estándares de códec de vídeo especifican cómo gestionar las múltiples imágenes de referencia.

En general, una lista de imágenes de referencia ("RPL") es una lista de imágenes de referencia utilizadas para la predicción con compensación de movimiento. En algunos formatos y estándares de códec de vídeo, un conjunto de imágenes de referencia ("RPS") es un conjunto de imágenes de referencia disponibles para su uso en predicción con compensación de movimiento en un momento dado, y una RPL son algunas de las imágenes de referencia en el RPS. Las imágenes de referencia en una RPL se abordan con índices de referencia. Un índice de referencia identifica una imagen de referencia en la RPL. Durante la codificación y decodificación, un RPS se puede actualizar para tener en cuenta las imágenes recién decodificadas y las imágenes más antiguas que ya no se utilizan como imágenes de referencia. Además, las imágenes de referencia dentro de una RPL se pueden reordenar de modo que las imágenes de referencia más comúnmente utilizadas se identifiquen con índices de referencia que sean más eficientes para señalizar. En algunos estándares de códec recientes, se construye una RPL durante la codificación y decodificación en base a la información disponible sobre el RPS, modificaciones según reglas y/o modificaciones señaladas en el flujo de bits. La señalización de modificaciones para una RPL puede consumir una cantidad significativa de bits.

### 40 Compendio

En resumen, la descripción detallada presenta innovaciones en la señalización de la información de modificación de la lista de imágenes de referencia ("RPL"). De manera más general, las innovaciones se refieren a diferentes formas de evitar la señalización de la información de modificación de la RPL cuando no se utilizaría o cuando pueden inferirse valores de dicha información. La invención se expone en el conjunto de reivindicaciones adjunto; los ejemplos adicionales denominados realizaciones en la descripción son ejemplos ilustrativos, no realizaciones reivindicadas en la presente solicitud.

Según un aspecto de las innovaciones descritas en este documento, un codificador de video evalúa una condición. Dependiendo de los resultados de la evaluación, el codificador señaliza de manera condicional en un flujo de bits una bandera que indica si una RPL se modifica según elementos de sintaxis señalados explícitamente en el flujo de bits. Un decodificador de video correspondiente evalúa una condición. Dependiendo de los resultados de la evaluación, el decodificador analiza condicionalmente a partir de un flujo de bits una bandera que indica si una RPL se modifica según elementos de sintaxis señalados explícitamente en el flujo de bits. En algunas implementaciones ejemplares, la RPL puede ser para un segmento predictivo ("P") o un segmento bi-predictivo ("B"). De manera alternativa, una estructura de sintaxis de nivel superior se señaliza/analiza de manera condicional en función de la evaluación de la condición.

En algunas implementaciones ejemplares, si no se modifica la RPL, se construye una RPL predeterminada basada en reglas sobre la construcción de RPL a partir de un RPS. Si se modifica la RPL, se construye una RPL de reemplazo basada en la información de modificación de la RPL señalizada que indica selecciones de imágenes de referencia del

RPS. Alternativamente, las modificaciones para reordenar una RPL predeterminada, agregar una imagen de referencia a la RPL predeterminada o eliminar una imagen de referencia de la RPL predeterminada se señalan de una manera más detallada para ajustar la RPL predeterminada.

5 Por ejemplo, la condición que se evalúa depende, al menos en parte, de una variable que indica un número total de imágenes de referencia. En algunas implementaciones ejemplares, la condición es si el valor de la variable es mayor que 1.

10 La condición se puede evaluar como parte del procesamiento de una estructura de modificación de RPL que incluye la bandera. O bien, la condición se puede evaluar como parte del procesamiento de una cabecera de segmento, en cuyo caso la estructura de modificación de RPL (incluida la bandera) se señala o analiza de manera condicional según los resultados de la evaluación.

15 Según otro aspecto de las innovaciones aquí descritas, un codificador de video evalúa otra condición. Dependiendo de los resultados de la evaluación, el codificador señala de manera condicional en un flujo de bits uno o más elementos de sintaxis para las entradas de la lista que indican cómo modificar una RPL (por ejemplo, reemplace la RPL, ajuste la RPL). Un decodificador de video correspondiente evalúa la condición. Dependiendo de los resultados de la evaluación, el decodificador analiza de manera condicional de un flujo de bits uno o más elementos de sintaxis para las entradas de la lista que indican cómo modificar una RPL (por ejemplo, reemplace la RPL, ajuste la RPL). En algunas implementaciones ejemplares, la RPL puede ser para un segmento P o un segmento B (con la evaluación de la condición y la señalización/análisis condicional repetidos para cada uno de las múltiples RPL para un segmento B). Por ejemplo, la otra condición depende al menos en parte de una variable que indica un número de imágenes de referencia totales, un número de imágenes de referencia activas para la RPL y/o si la predicción ponderada está inhabilitada. Se puede usar una lógica diferente para verificar si la predicción ponderada está deshabilitada dependiendo de si un segmento actual es un segmento P o un segmento B y/o dependiendo de qué RPL se esté señalando/analizando. En algunas implementaciones ejemplares, si (a) el número de imágenes de referencia totales es igual a 2 y (b) el número de imágenes de referencia activas para la RPL es igual a 1, entonces uno o más elementos de sintaxis para las entradas de la lista están ausentes del flujo de bits, y se infiere un valor para una de las entradas de la lista. Además, en algunas implementaciones ejemplares, si (c) el número de imágenes de referencia totales es igual a 2, (d) el número de imágenes de referencia activas para la RPL es igual a 2 y (e) la predicción ponderada está deshabilitada, entonces el uno o más elementos de sintaxis para las entradas de la lista están ausentes del flujo de bits y se infieren valores para dos de las entradas de la lista.

20 30 Según otro aspecto de las innovaciones descritas en este documento, un codificador de video evalúa otra condición. Dependiendo de los resultados de la evaluación, el codificador ajusta la señalización en un flujo de bits de uno o más elementos de sintaxis para las entradas de la lista que indican cómo modificar una RPL (por ejemplo, reemplace la RPL, ajuste la RPL). En particular, se ajusta la longitud (en bits) de al menos uno de los uno o más elementos de sintaxis. Un decodificador de video correspondiente evalúa la condición. Dependiendo de los resultados de la evaluación, el decodificador ajusta el análisis de un flujo de bits de uno o más elementos de sintaxis para las entradas de la lista que indican cómo modificar una RPL (nuevamente, donde la longitud (en bits) de al menos uno de los uno o más elementos de sintaxis es ajustada). Por ejemplo, la condición depende, al menos en parte, de si la predicción ponderada está deshabilitada. Se puede usar una lógica diferente para verificar si la predicción ponderada está deshabilitada dependiendo de si un segmento actual es un segmento P o un segmento B y/o dependiendo de qué RPL se esté señalando/analizando. En algunas implementaciones ejemplares, para un índice *i* para las entradas de la lista, si la predicción ponderada está deshabilitada, la longitud (en bits) de al menos uno de los elementos de sintaxis disminuye a medida que aumenta *i*. Por ejemplo, en algunas implementaciones ejemplares, si la predicción ponderada está deshabilitada, la longitud de un elemento de sintaxis dado para la entrada [i] de la lista es  $\text{Ceil}(\text{Log2}(\text{NumPocTotalAct}-i))$  bits. Por otro lado, si la predicción ponderada está habilitada, la longitud del elemento de sintaxis dado para la entrada [i] de la lista es  $\text{Ceil}(\text{Log2}(\text{NumPocTotalAct}))$  bits.

35 40 45 La codificación o decodificación se puede implementar como parte de un método, como parte de un dispositivo informático adaptado para realizar el método o como parte de un medio tangible legible por ordenador que almacena instrucciones ejecutables por ordenador para hacer que un dispositivo informático realice el método.

50 Los objetos, características y ventajas anteriores y otros de la invención resultarán más evidentes a partir de la siguiente descripción detallada, que procede con referencia a las figuras adjuntas.

#### Breve descripción de los dibujos

La Figura 1 es un diagrama de un sistema informático ejemplar en el que se pueden implementar algunas realizaciones descritas.

55 Las Figuras 2a y 2b son diagramas de entornos de red ejemplares en los que se pueden implementar algunas realizaciones descritas.

La Figura 3 es un diagrama de un sistema codificador ejemplar junto con el cual se pueden implementar algunas realizaciones descritas.

La Figura 4 es un diagrama de un sistema de decodificador ejemplar junto con el cual se pueden implementar algunas realizaciones descritas.

La Figura 5 es un diagrama que ilustra un codificador de video ejemplar junto con el cual se pueden implementar algunas realizaciones descritas.

5 La Figura 6 es un diagrama que ilustra un decodificador de video ejemplar junto con el cual se pueden implementar algunas realizaciones descritas.

La Figura 7a es una tabla que ilustra la señalización condicional de una bandera que indica si se modifica una RPL, según algunas implementaciones ejemplares.

10 Las Figuras 7b y 7c son tablas que ilustran la señalización condicional de una o más banderas que indican si se modifica una RPL, según otras implementaciones ejemplares.

Las Figuras 8 y 9 son tablas que ilustran la señalización condicional de elementos de sintaxis para entradas de lista que indican cómo modificar una RPL, según algunas implementaciones ejemplares.

Las Figuras 10 y 11 son diagramas de flujo que ilustran técnicas generalizadas para señalización y análisis sintáctico condicional, respectivamente, de una bandera que indica si se modifica una RPL.

15 Las Figuras 12 y 13 son diagramas de flujo que ilustran técnicas generalizadas para señalización condicional y análisis sintáctico, respectivamente, de elementos de sintaxis para entradas de lista que indican cómo modificar una RPL.

Las Figuras 14 y 15 son diagramas de flujo que ilustran técnicas generalizadas para ajustar la señalización y el análisis sintáctico, respectivamente, de elementos de sintaxis para entradas de lista que indican cómo modificar una RPL.

### Descripción detallada

20 La descripción detallada presenta innovaciones en la señalización de la información de modificación de la lista de imágenes de referencia ("RPL"). Estas innovaciones pueden ayudar a evitar la señalización de información de modificación de RPL cuando no se usaría o cuando se pueden inferir valores de dicha información.

En algunos estándares de códec recientes, un conjunto de imágenes de referencia ("RPS") es un conjunto de imágenes de referencia disponibles para su uso en la predicción con compensación de movimiento, y una RPL se construye a partir del RPS. Para el proceso de decodificación de un segmento predictivo ("P"), hay una RPL, que se llama RPL 0. Para el proceso de decodificación de un segmento bi-predictivo ("B"), hay dos RPL, que se denominan RPL 0 y RPL 1. Al comienzo del proceso de decodificación para un segmento P, RPL 0 se deriva de la información disponible sobre RPL 0 (como el conjunto de imágenes de referencia disponibles en el decodificador para decodificar la imagen actual), modificaciones según reglas y/o modificaciones señaladas en el flujo de bits. De manera similar, al comienzo del proceso de decodificación para un segmento B, RPL 0 y RPL 1 se derivan de la información disponible sobre RPL 0 y la información disponible sobre RPL 1 (como el conjunto de imágenes de referencia disponibles en el decodificador para decodificar la corriente imagen), modificaciones según reglas y/o modificaciones señaladas en el flujo de bits. Más generalmente, una RPL se construye durante la codificación y decodificación basándose en la información disponible sobre la RPL, modificaciones según reglas y/o modificaciones señaladas en el flujo de bits. La señalización de modificaciones para una RPL puede consumir una cantidad significativa de bits. Para algunos estándares de códec recientes, existen ineficiencias en la forma en que se señaliza la información de modificación de RPL.

40 La descripción detallada presenta varias innovaciones en el área de señalización de información de modificación de RPL. En algunas situaciones, estas innovaciones dan como resultado una señalización más eficiente de los elementos de sintaxis para la información de modificación de RPL. Por ejemplo, la descripción detallada describe la señalización condicional de elementos de sintaxis para entradas de lista que indican cómo modificar una RPL. La descripción detallada también describe formas de utilizar menos bits para señalar dichos elementos de sintaxis. Como otro ejemplo, la descripción detallada describe la señalización condicional de una bandera que indica si se modifica una RPL.

45 En algunas implementaciones ejemplares, si no se modifica la RPL, se construye una RPL predeterminada según un enfoque "implícito" mediante el uso de reglas sobre la construcción de RPL a partir de un RPS. Si se modifica la RPL, se construye una RPL de reemplazo según un enfoque de señalización "explícito" mediante el uso de información de modificación de RPL señalizada que indica selecciones de imágenes de referencia del RPS. Alternativamente, las modificaciones para reordenar, agregar una imagen de referencia o eliminar una imagen de referencia de una RPL predeterminada se pueden señalar de una manera más detallada como cambios específicos en relación con la RPL predeterminada.

50 Algunas de las innovaciones descritas en este documento se ilustran con referencia a elementos de sintaxis y operaciones específicas del estándar HEVC. Por ejemplo, se hace referencia a la versión preliminar JCTVC-I1003 del estándar HEVC - "High efficiency video coding (HEVC) text specification draft 7", JCTVC-I1003\_d5, 9<sup>a</sup> reunión del

Equipo Colaborativo Conjunto sobre Codificación de Video ("JCT-VC"), Ginebra, abril del 2012. Consulte también la versión preliminar titulada "High efficiency video coding (HEVC) text specification draft 9", JCTVC-K1003\_d11, 11<sup>a</sup> reunión del JCT-VC, Shanghai, octubre del 2012. Las innovaciones aquí descritas también se pueden implementar para otros estándares o formatos.

- 5 Más generalmente, son posibles varias alternativas a los ejemplos descritos en este documento. Por ejemplo, algunos de los métodos descritos en el presente documento pueden modificarse mediante el cambio del orden de los actos del método descritos, al dividir, repetir u omitir ciertos actos del método, etc. Los diversos aspectos de la tecnología descrita se pueden usar en combinación o por separado. Diferentes realizaciones utilizan una o más de las innovaciones descritas. Algunas de las innovaciones descritas en este documento abordan uno o más de los problemas señalados en segundo plano. Normalmente, una técnica/herramienta determinada no resuelve todos estos problemas.

I. Sistemas informáticos ejemplares.

- 15 La Figura 1 ilustra un ejemplo generalizado de un sistema (100) informático adecuado en el que se pueden implementar varias de las innovaciones descritas. El sistema (100) informático no pretende sugerir ninguna limitación en cuanto al alcance de uso o funcionalidad, ya que las innovaciones pueden implementarse en diversos sistemas informáticos de propósito general o especial.

- 20 Con referencia a la Figura 1, el sistema (100) informático incluye una o más unidades (110, 115) de procesamiento y memoria (120, 125). En la Figura 1, esta configuración (130) más básica se incluye dentro de una línea discontinua. Las unidades (110, 115) de procesamiento ejecutan instrucciones ejecutables por ordenador. Una unidad de procesamiento puede ser una unidad de procesamiento central de propósito general ("CPU"), un procesador en un circuito integrado de aplicación específica ("ASIC") o cualquier otro tipo de procesador. En un sistema de multiprocesamiento, varias unidades de procesamiento ejecutan instrucciones ejecutables por ordenador para aumentar la potencia de procesamiento. Por ejemplo, la Figura 1 muestra una unidad (110) central de procesamiento así como una unidad (115) de procesamiento de gráficos o una unidad de coprocesamiento. La memoria (120, 125) tangible puede ser una memoria volátil (por ejemplo, registros, caché, RAM), memoria no volátil (por ejemplo, ROM, EEPROM, memoria flash, etc.), o alguna combinación de las dos, accesible por la(s) unidad(es) de procesamiento. La memoria (120, 125) almacena software (180) que implementa una o más innovaciones para señalizar la información de modificación de RPL, en forma de instrucciones ejecutables por ordenador adecuadas para su ejecución por la(s) unidad(es) de procesamiento.

- 30 Un sistema informático puede tener características adicionales. Por ejemplo, el sistema (100) informático incluye almacenamiento (140), uno o más dispositivos (150) de entrada, uno o más dispositivos (160) de salida y una o más conexiones (170) de comunicación. Un mecanismo de interconexión (no mostrado) tal como un bus, controlador o red interconecta los componentes del sistema (100) informático. Normalmente, el software del sistema operativo (no mostrado) proporciona un entorno operativo para otro software que se ejecuta en el sistema (100) informático y coordina las actividades de los componentes del sistema (100) informático.

- 35 El almacenamiento (140) tangible puede ser extraíble o no extraíble, e incluye discos magnéticos, cintas o casetes magnéticos, CD-ROM, DVD o cualquier otro medio que se pueda utilizar para almacenar información y al que se pueda acceder dentro del sistema (100) informático. El almacenamiento (140) almacena instrucciones para el software (180) que implementa una o más innovaciones para señalizar la información de modificación de RPL.

- 40 El/los dispositivo(s) (150) de entrada puede(n) ser un dispositivo de entrada táctil como un teclado, mouse, bolígrafo o trackball, un dispositivo de entrada de voz, un dispositivo de escaneo u otro dispositivo que proporcione entrada al sistema (100) informático. Para la codificación de video, el/los dispositivo(s) (150) de entrada puede(n) ser una cámara, tarjeta de video, tarjeta sintonizadora de TV o dispositivo similar que acepta entrada de video en forma analógica o digital, o un CD-ROM o CD-RW que lee muestras de video en el sistema (100) informático. El/los dispositivo(s) (160) de salida pueden ser una pantalla, una impresora, un altavoz, una grabadora de CD u otro dispositivo que proporcione salida desde el sistema (100) informático.

- 45 La(s) conexión/conexiones (170) de comunicación permite(n) la comunicación a través de un medio de comunicación a otra entidad informática. El medio de comunicación transmite información tal como instrucciones ejecutables por ordenador, entrada o salida de audio o video, u otros datos en una señal de datos modulada. Una señal de datos modulada es una señal que tiene una o más de sus características establecidas o cambiadas de tal manera que codifica información en la señal. A modo de ejemplo, y no de limitación, los medios de comunicación pueden utilizar una portadora eléctrica, óptica, RF u otra.

- 50 Las innovaciones se pueden describir en el contexto general de los medios legibles por ordenador. Los medios legibles por ordenador son cualquier medio tangible disponible al que se puede acceder dentro de un entorno informático. A modo de ejemplo, y no de limitación, con el sistema (100) informático, los medios legibles por ordenador incluyen memoria (120, 125), almacenamiento (140) y combinaciones de cualquiera de los anteriores.

- 55 Las innovaciones se pueden describir en el contexto general de instrucciones ejecutables por ordenador, como las incluidas en los módulos del programa, que se ejecutan en un sistema informático en un procesador objetivo real o

virtual. Generalmente, los módulos de programa incluyen rutinas, programas, bibliotecas, objetos, clases, componentes, estructuras de datos, etc. que realizan tareas particulares o implementan tipos de datos abstractos particulares. La funcionalidad de los módulos de programa puede combinarse o dividirse entre módulos de programa según se desee en varias realizaciones. Las instrucciones ejecutables por ordenador para módulos de programa pueden ejecutarse dentro de un sistema informático local o distribuido.

Los términos "sistema" y "dispositivo" se usan indistintamente en el presente documento. A menos que el contexto indique claramente lo contrario, ninguno de los términos implica ninguna limitación en un tipo de sistema informático o dispositivo informático. En general, un sistema informático o dispositivo informático puede ser local o distribuido, y puede incluir cualquier combinación de hardware de propósito especial y/o hardware de propósito general con software que implemente la funcionalidad aquí descrita.

Los métodos descritos también se pueden implementar mediante el uso de hardware informático especializado configurado para realizar cualquiera de los métodos descritos. Por ejemplo, los métodos descritos pueden implementarse mediante un circuito integrado (por ejemplo, un circuito integrado específico de la aplicación ("ASIC") (como una unidad de proceso de señal digital ASIC ("DSP"), una unidad de procesamiento de gráficos ("GPU") o un dispositivo lógico programable ("PLD"), como un matriz de puertas programables en campo ("FPGA")) especialmente diseñada o configurada para implementar cualquiera de los métodos descritos.

En aras de la presentación, la descripción detallada utiliza términos como "determinar" y "utilizar" para describir las operaciones del ordenador en un sistema informático. Estos términos son abstracciones de alto nivel para operaciones realizadas por un ordenador y no deben confundirse con actos realizados por un ser humano. Las operaciones informáticas reales correspondientes a estos términos varían según la implementación.

## II. Entornos de red ejemplares.

Las Figuras 2a y 2b muestran entornos (201, 202) de red ejemplares que incluyen codificadores (220) de video y decodificadores (270) de video. Los codificadores (220) y decodificadores (270) están conectados a través de una red (250) mediante el uso de un protocolo de comunicación apropiado. La red (250) puede incluir Internet u otra red informática.

En el entorno (201) de red mostrado en la Figura 2a, cada herramienta (210) de comunicación en tiempo real ("RTC") incluye tanto un codificador (220) como un decodificador (270) para comunicación bidireccional. Un codificador (220) determinado puede producir una salida que cumpla con el estándar SMPTE 421M, la norma ISO-IEC 14496-10 (también conocido como H.264 o AVC), el estándar HEVC, otro estándar o un formato propietario, con el decodificador (270) correspondiente que acepta datos codificados del codificador (220). La comunicación bidireccional puede ser parte de una videoconferencia, videollamada u otro escenario de comunicación entre dos partes. Aunque el entorno (201) de red en la Figura 2a incluye dos herramientas (210) de comunicación en tiempo real, el entorno (201) de red puede incluir en su lugar tres o más herramientas (210) de comunicación en tiempo real que participan en la comunicación multipartita.

Una herramienta (210) de comunicación en tiempo real gestiona la codificación mediante un codificador (220). La Figura 3 muestra un sistema (300) codificador ejemplar que se puede incluir en la herramienta (210) de comunicación en tiempo real. Alternativamente, la herramienta (210) de comunicación en tiempo real usa otro sistema codificador. Una herramienta (210) de comunicación en tiempo real también gestiona la decodificación mediante un decodificador (270). La Figura 4 muestra un sistema (400) de decodificador ejemplar, que puede incluirse en la herramienta (210) de comunicación en tiempo real. De manera alternativa, la herramienta (210) de comunicación en tiempo real usa otro sistema decodificador.

En el entorno (202) de red que se muestra en la Figura 2b, una herramienta (212) de codificación incluye un codificador (220) que codifica vídeo para su entrega a múltiples herramientas (214) de reproducción, que incluyen decodificadores (270). La comunicación unidireccional se puede proporcionar para un sistema de videovigilancia, un sistema de monitorización de cámara web, una presentación de conferencia de escritorio remoto u otro escenario en el que el video se codifica y envía desde una ubicación a una o más ubicaciones. Aunque el entorno (202) de red en la Figura 2b incluye dos herramientas (214) de reproducción, el entorno (202) de red puede incluir más o menos herramientas (214) de reproducción. En general, una herramienta (214) de reproducción se comunica con la herramienta (212) de codificación para determinar un flujo de vídeo para que lo reciba la herramienta (214) de reproducción. La herramienta (214) de reproducción recibe el flujo, almacena los datos codificados recibidos durante un período apropiado y comienza a decodificar y reproducir.

La Figura 3 muestra un sistema (300) codificador ejemplar que se puede incluir en la herramienta (212) de codificación. De manera alternativa, la herramienta (212) de codificación usa otro sistema de codificación. La herramienta (212) de codificación también puede incluir lógica de controlador del lado del servidor para gestionar conexiones con una o más herramientas (214) de reproducción. La Figura 4 muestra un sistema (400) de decodificador ejemplar, que puede incluirse en la herramienta (214) de reproducción. De manera alternativa, la herramienta (214) de reproducción usa otro sistema decodificador. Una herramienta (214) de reproducción también puede incluir lógica de controlador del lado del cliente para gestionar conexiones con la herramienta (212) de codificación.

## III. Sistemas codificadores ejemplares.

La Figura 3 es un diagrama de bloques de un sistema (300) codificador ejemplar junto con el cual se pueden implementar algunas realizaciones descritas. El sistema (300) codificador puede ser una herramienta de codificación de propósito general capaz de operar en cualquiera de los múltiples modos de codificación, como un modo de codificación de baja latencia para comunicación en tiempo real, modo de transcodificación y modo de codificación regular para reproducción de medios desde un archivo o flujo, o puede ser una herramienta de codificación de propósito especial adaptada para tal modo de codificación. El sistema (300) codificador se puede implementar como un módulo del sistema operativo, como parte de una biblioteca de aplicaciones o como una aplicación independiente. En general, el sistema (300) codificador recibe una secuencia de fotogramas (311) de video fuente desde una fuente (310) de video y produce datos codificados como salida a un canal (390). La salida de datos codificados al canal puede incluir elementos de sintaxis que indican información de modificación de RPL.

La fuente (310) de video puede ser una cámara, tarjeta sintonizadora, medio de almacenamiento u otra fuente de video digital. La fuente (310) de video produce una secuencia de fotogramas de video a una velocidad de fotograma de, por ejemplo, 30 fotogramas por segundo. Como se usa en este documento, el término "fotograma" generalmente se refiere a datos de imagen de fuente, codificados o reconstruidos. Para video progresivo, un fotograma es un fotograma de video progresivo. Para el video entrelazado, en las formas de realización ejemplares, un fotograma de video entrelazado se desentrelaza antes de la codificación. Alternativamente, dos campos de video entrelazados complementarios se codifican como un fotograma de video entrelazado o campos separados. Además de indicar un fotograma de video progresivo, el término "fotograma" o "imagen" puede indicar un único campo de video no emparejado, un par de campos de video complementarios, un plano de objeto de video que representa un objeto de video en un momento determinado o una región de interés en una imagen más grande. El plano o región del objeto de video puede ser parte de una imagen más grande que incluye múltiples objetos o regiones de una escena.

Un fotograma (311) fuente que llega se almacena en un área (320) de almacenamiento de memoria temporal de fotogramas fuente que incluye múltiples áreas (321, 322, ..., 32n) de almacenamiento de memoria intermedia de fotogramas. Una memoria intermedia (321, 322, etc.) de fotogramas contiene un fotograma fuente en el área (320) de almacenamiento de fotogramas fuente. Después de que uno o más de los fotogramas (311) fuente se hayan almacenado en memorias intermedias (321, 322, etc.) de fotogramas, un selector (330) de fotogramas selecciona periódicamente un fotograma fuente individual del área (320) de almacenamiento de fotogramas fuente. El orden en el que los fotogramas son seleccionados por el selector (330) de fotogramas para la entrada al codificador (340) puede diferir del orden en el que los fotogramas son producidos por la fuente (310) de video, por ejemplo, un fotograma puede estar adelante en orden, para facilitar la predicción temporalmente hacia atrás. Antes del codificador (340), el sistema (300) codificador puede incluir un preprocesador (no mostrado) que realiza el preprocesamiento (por ejemplo, filtrado) de los fotogramas antes de la codificación. El preprocesamiento también puede incluir la conversión del espacio de color en componentes primarios y secundarios para la codificación.

El codificador (340) codifica el fotograma (331) seleccionado para producir un fotograma (341) codificado y también produce señales (342) de operación de control de gestión de memoria ("MMCO") o información de conjunto de imágenes de referencia ("RPS"). Si el fotograma actual no es el primer fotograma que ha sido codificado, al realizar su proceso de codificación, el codificador (340) puede usar uno o más fotogramas (369) previamente codificados/decodificados que han sido almacenados en un área (360) de almacenamiento de memoria temporal de fotogramas decodificados. Tales fotogramas (369) decodificados almacenados se utilizan como imágenes de referencia para la predicción entre fotogramas del contenido del fotograma (331) fuente actual. Generalmente, el codificador (340) incluye múltiples módulos de codificación que realizan tareas de codificación tales como estimación y compensación de movimiento, transformaciones de frecuencia, cuantificación y codificación de entropía. Las operaciones exactas realizadas por el codificador (340) pueden variar según el formato de compresión. El formato de los datos codificados de salida puede ser un formato de video de Windows Media, formato VC-1, formato MPEG-x (p. Ej., MPEG-1, MPEG-2 o MPEG-4), formato H.26x (p. Ej., H.261, H.262, H.263, H.264), formato HEVC u otro formato.

Por ejemplo, dentro del codificador (340), un fotograma predicho intercodificado se representa en términos de predicción a partir de fotogramas de referencia, que son ejemplos de imágenes de referencia. Un estimador de movimiento estima el movimiento de bloques u otros conjuntos de muestras de un fotograma (341) fuente con respecto a uno o más fotogramas (369) de referencia. Cuando se utilizan múltiples fotogramas de referencia, los múltiples fotogramas de referencia pueden ser de diferentes direcciones temporales o de la misma dirección temporal. Los fotogramas de referencia (imágenes de referencia) pueden ser parte de una o más RPL, con índices de referencia que abordan las imágenes de referencia en la(s) RPL. La(s) RPL se construyen durante la codificación para que se agreguen nuevas imágenes de referencia cuando sea apropiado, las imágenes de referencia más antiguas que ya no se usan para la compensación de movimiento se eliminan cuando corresponde, y las imágenes de referencia se reordenan cuando corresponde. En algunas implementaciones, por ejemplo, al codificar una imagen actual, el codificador (340) determina un RPS que incluye imágenes de referencia en el área (360) de almacenamiento de fotogramas decodificados, luego crea una o más RPL para codificar un segmento dado de la imagen actual. Se puede crear una RPL mediante la aplicación de reglas sobre la selección de imágenes de referencia disponibles del RPS (enfoque implícito), en cuyo caso la información de modificación de la RPL no se señaliza explícitamente en el flujo de bits. O bien, la RPL se puede crear mediante la selección de imágenes de referencia específicas disponibles en el RPS, donde las imágenes de referencia que se seleccionan se indicarán en la información de modificación de RPL.

que se señala en el flujo de bits. En comparación con una RPL que se construiría mediante reglas del enfoque implícito, la información de modificación de la RPL puede especificar una RPL de reemplazo como una lista de imágenes de referencia en el RPS. Alternativamente, la información de modificación de RPL puede, de una manera más detallada, especificar la eliminación de una o más imágenes de referencia, la adición de una o más imágenes de referencia y/o el reordenamiento de las imágenes de referencia en la RPL construida por reglas del enfoque implícito.

Al codificar un fotograma intercodificado, el codificador (340) puede evaluar los resultados de la compensación de movimiento para los que no se modifica una RPL según los elementos de sintaxis señalados explícitamente en el flujo de bits, y también evaluar los resultados de la compensación de movimiento para los que la RPL es modificada según elementos de sintaxis señalados explícitamente en el flujo de bits (o resultados de múltiples formas diferentes de modificar la RPL). El codificador (340) puede decidir utilizar la RPL predeterminada (sin información de modificación de RPL señalizada en el flujo de bits) o una RPL modificada (con información de modificación de RPL señalizada en el flujo de bits). Cuando se modifica la RPL (por ejemplo, reemplazada, ajustada), en comparación con la RPL predeterminada, el codificador (340) puede realizar una o más de (a) reordenar imágenes de referencia para un direccionamiento más eficiente con índices de referencia, (b) eliminar imágenes de referencia basadas al menos en parte en frecuencia de uso durante la codificación, y (c) agregar imágenes de referencia basadas al menos en parte en la frecuencia de uso durante la codificación. Por ejemplo, el codificador (340) puede decidir eliminar una imagen de referencia dada de la RPL después de que la utilización de la imagen de referencia para la compensación de movimiento caiga por debajo de una cantidad umbral y/o según otros criterios. Como otro ejemplo, el codificador (340) puede decidir agregar una imagen de referencia dada a la RPL si la utilización de la imagen de referencia para la compensación de movimiento está por encima de una cantidad umbral y/o según otros criterios. Como otro ejemplo, el codificador (340) puede decidir cómo reordenar las imágenes de referencia en la RPL basándose en la frecuencia de utilización de las respectivas imágenes de referencia y/o según otros criterios.

El estimador de movimiento genera información de movimiento, como información de vector de movimiento, que está codificada por entropía. Un compensador de movimiento aplica vectores de movimiento a imágenes de referencia para determinar valores de predicción compensados por movimiento. El codificador determina las diferencias (si las hay) entre los valores de predicción con compensación de movimiento de un bloque y los valores originales correspondientes. Estos valores residuales de predicción se codifican adicionalmente mediante el uso de una transformación de frecuencia, cuantificación y codificación de entropía. De manera similar, para la predicción interna, el codificador (340) puede determinar valores de predicción interna para un bloque, determinar valores residuales de predicción y codificar los valores residuales de predicción (con una codificación de transformación de frecuencia, cuantificación y entropía). En particular, el codificador de entropía del codificador (340) comprime valores de coeficientes de transformación cuantificados así como cierta información lateral (por ejemplo, información de vector de movimiento, valores de QP, decisiones de modo, opciones de parámetros, índices de referencia, información de modificación de RPL). Las técnicas típicas de codificación de entropía incluyen codificación Exp-Golomb, codificación aritmética, codificación diferencial, codificación Huffman, codificación de longitud de ejecución, codificación de longitud variable a longitud variable ("V2V"), longitud variable a longitud fija ("V2F"), codificación LZ, codificación de diccionario, codificación de entropía de partición de intervalo de probabilidad ("PIPE") y combinaciones de los anteriores. El codificador de entropía puede usar diferentes técnicas de codificación para diferentes tipos de información y puede elegir entre múltiples tablas de códigos dentro de una técnica de codificación particular.

Los fotogramas (341) codificados y la información (342) MMCO/RPS son procesados por un emulador (350) del proceso de decodificación. El emulador (350) del proceso de decodificación implementa algunas de las funciones de un decodificador, por ejemplo, tareas de decodificación para reconstruir imágenes de referencia que son utilizadas por el codificador (340) en compensación de movimiento. El emulador (350) del proceso de decodificación usa la información (342) MMCO/RPS para determinar si un fotograma (341) codificado dado necesita ser reconstruido y almacenado para su uso como imagen de referencia en la predicción entre fotogramas subsiguientes a codificar. Si la información (342) MMCO/RPS indica que un fotograma (341) codificado necesita ser almacenado, el emulador (350) de proceso de decodificación modela el proceso de decodificación que sería realizado por un decodificador que recibe el fotograma (341) codificado y produce un fotograma (351) decodificado correspondiente. Al hacerlo, cuando el codificador (340) ha utilizado fotograma(s) (369) decodificado(s) que se ha(n) almacenado en el área (360) de almacenamiento de fotogramas decodificadas, el emulador (350) del proceso de decodificación también usa el/los fotograma(s) (369) decodificado(s) del área (360) de almacenamiento como parte del proceso de decodificación.

El área (360) de almacenamiento de memoria temporal de fotogramas decodificados incluye múltiples áreas (361, 362, ..., 36n) de almacenamiento de memoria intermedia de fotogramas. El emulador (350) del proceso de decodificación utiliza la información (342) MMCO/RPS para gestionar el contenido del área (360) de almacenamiento con el fin de identificar cualquier memoria intermedia (361, 362, etc.) de fotogramas con fotogramas que ya no son necesarias por codificador (340) para utilizar como imágenes de referencia. Después de modelar el proceso de decodificación, el emulador (350) del proceso de decodificación almacena un fotograma (351) recién decodificado en una memoria intermedia (361, 362, etc.) de fotogramas que ha sido identificada de esta manera.

Los fotogramas (341) codificados y la información (342) MMCO/RPS también se almacenan temporalmente en un área (370) de datos codificados temporal. Los datos codificados que se agregan en el área (370) de datos codificados pueden contener, como parte de la sintaxis de un flujo de bits de vídeo codificado elemental, elementos de sintaxis

que indican información de modificación de RPL. Los datos codificados que se agregan en el área (370) de datos codificados también pueden incluir metadatos de medios relacionados con los datos de video codificados (por ejemplo, como uno o más parámetros en uno o más mensajes de información de mejora suplementaria ("SEI") o mensajes de información de usabilidad de vídeo ("VUI")).

- 5 Los datos (371) agregados del área (370) de datos codificados temporalmente son procesados por un codificador (380) de canal. El codificador (380) de canal puede empaquetar los datos agregados para su transmisión como un flujo de medios (por ejemplo, según un formato contenedor de medios como ISO/IEC 14496-12), en cuyo caso el codificador (380) de canal puede agregar elementos de sintaxis como parte de la sintaxis del flujo de transmisión de medios. O, el codificador (380) de canal puede organizar los datos agregados para almacenarlos como un archivo (por ejemplo, según un formato de contenedor de medios como ISO/IEC 14496-12), en cuyo caso el codificador (380) de canal puede agregar elementos de sintaxis como parte de la sintaxis del archivo de almacenamiento de medios. O, de forma más general, el codificador (380) de canal puede implementar uno o más protocolos de multiplexación de sistemas de medios o protocolos de transporte, en cuyo caso el codificador (380) de canal puede añadir elementos de sintaxis como parte de la sintaxis del/de los protocolo(s). El codificador (380) de canal proporciona salida a un canal (390), que representa almacenamiento, una conexión de comunicaciones u otro canal para la salida.

#### IV. Sistemas decodificadores ejemplares.

La Figura 4 es un diagrama de bloques de un sistema (400) decodificador ejemplar junto con el cual se pueden implementar algunas realizaciones descritas. El sistema (400) decodificador puede ser una herramienta de decodificación de propósito general capaz de operar en cualquiera de los múltiples modos de decodificación, como un modo de decodificación de baja latencia para comunicación en tiempo real y un modo de decodificación regular para reproducción de medios desde un archivo o flujo, o puede ser una herramienta de decodificación de propósito especial adaptada para tal modo de decodificación. El sistema (400) decodificador se puede implementar como un módulo del sistema operativo, como parte de una biblioteca de aplicaciones o como una aplicación independiente. En general, el sistema (400) decodificador recibe datos codificados de un canal (410) y produce fotogramas reconstruidos como salida para un destino (490) de salida. Los datos codificados pueden incluir elementos de sintaxis que indican información de modificación de RPL.

El sistema (400) decodificador incluye un canal (410), que puede representar almacenamiento, una conexión de comunicaciones u otro canal para datos codificados como entrada. El canal (410) produce datos codificados que han sido codificados por canal. Un decodificador (420) de canal puede procesar los datos codificados. Por ejemplo, el decodificador (420) de canal desempaquetar los datos que se han agregado para su transmisión como un flujo de medios (por ejemplo, según un formato contenedor de medios como ISO/IEC 14496-12), en cuyo caso el decodificador (420) de canal puede analizar elementos sintácticos agregados como parte de la sintaxis del flujo de transmisión de medios. O, el decodificador (420) de canal separa los datos de video codificados que se han agregado para su almacenamiento como un archivo (por ejemplo, según un formato contenedor de medios como ISO/IEC 14496-12), en cuyo caso el decodificador (420) de canal puede analizar los elementos sintácticos agregados como parte de la sintaxis del archivo de almacenamiento de medios. O, más generalmente, el decodificador (420) de canal puede implementar uno o más protocolos de demultiplexación del sistema de medios o protocolos de transporte, en cuyo caso el decodificador (420) de canal puede analizar elementos sintácticos añadidos como parte de la sintaxis del/de los protocolo(s).

40 Los datos (421) codificados que salen del decodificador (420) de canal se almacenan en un área (430) de datos codificados temporal hasta que se haya recibido una cantidad suficiente de tales datos. Los datos (421) codificados incluyen fotogramas (431) codificados e información (432) MMCO/RPS. Los datos (421) codificados en el área (430) de datos codificados pueden contener, como parte de la sintaxis de un flujo de bits de video codificado elemental, elementos de sintaxis que indican información de modificación de RPL. Los datos (421) codificados en el área (430) de datos codificados también pueden incluir metadatos de medios relacionados con los datos de video codificados (por ejemplo, como uno o más parámetros en uno o más mensajes SEI o mensajes VUI). En general, el área (430) de datos codificados almacena temporalmente datos (421) codificados hasta que el decodificador (450) utiliza dichos datos (421) codificados. En ese punto, los datos codificados para un fotograma (431) codificado y la información (432) MMCO/RPS se transfieren desde el área (430) de datos codificados al decodificador (450). A medida que continúa la decodificación, se añaden nuevos datos codificados al área (430) de datos codificados y los datos codificados más antiguos que quedan en el área (430) de datos codificados se transfieren al decodificador (450).

45 El decodificador (450) decodifica periódicamente un fotograma (431) codificado para producir un fotograma (451) decodificado correspondiente. Según sea apropiado, al realizar su proceso de decodificación, el decodificador (450) puede usar uno o más fotogramas (469) previamente decodificados como fotogramas de referencia (imágenes de referencia) para la predicción entre fotogramas. El decodificador (450) lee dichos fotogramas (469) decodificadas previamente de un área (460) de almacenamiento de memoria temporal de fotogramas decodificados. Generalmente, el decodificador (450) incluye múltiples módulos de decodificación que realizan tareas de decodificación tales como decodificación de entropía, cuantificación inversa, transformaciones de frecuencia inversa y compensación de movimiento (que pueden crear RPL mediante el uso de información de modificación de RPL). Las operaciones exactas realizadas por el decodificador (450) pueden variar según el formato de compresión.

Por ejemplo, el decodificador (450) recibe datos codificados para un fotograma comprimido o secuencia de fotogramas y produce una salida que incluye el fotograma (451) decodificado. En el decodificador (450), una memoria intermedia recibe datos codificados para un fotograma comprimido y hace que los datos codificados recibidos estén disponibles para un decodificador de entropía. El decodificador de entropía decodifica en entropía datos cuantificados codificados en entropía, así como información lateral codificada en entropía (incluidos índices de referencia, información de modificación de RPL, etc.), mediante la aplicación de normalmente la codificación inversa de entropía realizada en el codificador. El decodificador construye una o más RPL para imágenes de referencia, con índices de referencia que se refieren a las imágenes de referencia en la(s) RPL. La(s) RPL están construidas para que se agreguen nuevas imágenes de referencia cuando sea apropiado, las imágenes de referencia más antiguas que ya no se utilizan para la compensación de movimiento se eliminan cuando corresponde, y las imágenes de referencia se reordenan cuando corresponde. En algunas implementaciones, por ejemplo, al decodificar una imagen actual, el decodificador (450) determina un RPS que incluye imágenes de referencia en el área (460) de almacenamiento de fotogramas decodificados, luego crea una o más RPL para decodificar un segmento dado de la imagen actual. Se puede crear una RPL mediante la aplicación de reglas sobre la selección de imágenes de referencia disponibles en el RPS, en cuyo caso la información de modificación de la RPL no se analiza a partir del flujo de bits. O bien, la RPL se puede crear mediante la selección de imágenes de referencia específicas disponibles en el RPS, donde las imágenes de referencia que se seleccionan se indican en la información de modificación de RPL que se analiza a partir del flujo de bits. En comparación con una RPL que se construiría mediante reglas del enfoque implícito, la información de modificación de la RPL puede especificar una RPL de reemplazo como una lista de imágenes de referencia en el RPS. Alternativamente, la información de modificación de RPL puede, de una manera más detallada, especificar la eliminación de una o más imágenes de referencia, la adición de una o más imágenes de referencia y/o el reordenamiento de las imágenes de referencia en la RPL construida por reglas del enfoque implícito.

Un compensador de movimiento aplica información de movimiento a una o más imágenes de referencia para formar predicciones compensadas por movimiento de sub-bloques y/o bloques (generalmente, bloques) del fotograma que se está reconstruyendo. Un módulo de intra predicción puede predecir espacialmente valores de muestra de un bloque actual a partir de valores de muestra vecinos previamente reconstruidos. El decodificador (450) también reconstruye los residuos de predicción. Un cuantificador inverso cuantifica los datos decodificados por entropía. Un transformador de frecuencia inversa convierte los datos de dominio de frecuencia reconstruidos en información de dominio espacial. Para un fotograma predicho, el decodificador (450) combina residuos de predicción reconstruidos con predicciones compensadas por movimiento para formar un fotograma reconstruido. El decodificador (450) puede combinar de manera similar residuos de predicción con predicciones espaciales de intra predicción. Un bucle de compensación de movimiento en el decodificador (450) de vídeo incluye un filtro de desbloqueo adaptativo para suavizar las discontinuidades a través de filas y/o columnas de límites de bloque en el fotograma (451) decodificado.

El área (460) de almacenamiento de memoria temporal de fotogramas decodificadas incluye múltiples áreas (461, 462, ..., 46n) de almacenamiento de memoria intermedia de fotogramas. El área (460) de almacenamiento de fotogramas decodificadas es un ejemplo de un DPB. El decodificador (450) usa la información (432) MMCO/RPS para identificar una memoria intermedia (461, 462, etc.) de fotogramas en la que puede almacenar un fotograma (451) decodificado. El decodificador (450) almacena el fotograma (451) decodificado en esa memoria intermedia de fotogramas.

Un secuenciador (480) de salida usa la información (432) MMCO/RPS para identificar cuándo está disponible el siguiente fotograma a producir en el orden de salida en el área (460) de almacenamiento de fotogramas decodificados. Cuando el siguiente fotograma (481) que se va a producir en el orden de salida está disponible en el área (460) de almacenamiento de fotogramas decodificados, el secuenciador (480) de salida lo lee y lo envía al destino (490) de salida (por ejemplo, mostrar). En general, el orden en el que los fotogramas son emitidos desde el área (460) de almacenamiento de fotogramas decodificados por el secuenciador (480) de salida puede diferir del orden en el que los fotogramas son decodificados por el decodificador (450).

#### V. Codificadores de video ejemplares.

La Figura 5 es un diagrama de bloques de un codificador (500) de video generalizado junto con el cual se pueden implementar algunas realizaciones descritas. El codificador (500) recibe una secuencia de fotogramas de vídeo que incluye un fotograma (505) actual y produce datos (595) codificados como salida.

El codificador (500) está basado en bloques y usa un formato de bloque que depende de la implementación. Los bloques pueden subdividirse aún más en diferentes etapas, por ejemplo, en las etapas de codificación de entropía y transformación de frecuencia. Por ejemplo, un fotograma se puede dividir en bloques de 64x64, bloques de 32x32 o bloques de 16x16, que a su vez se pueden dividir en bloques más pequeños y sub-bloques de valores de píxeles para codificar y decodificar.

El sistema (500) codificador comprime los fotogramas pronosticados y los fotogramas intracodificados. En aras de la presentación, la Figura 5 muestra una "ruta interna" a través del codificador (500) para la codificación intrafotograma y una "ruta intermedia" para la codificación entre fotogramas. Muchos de los componentes del codificador (500) se utilizan tanto para la codificación dentro del fotograma como para la codificación entre fotogramas. Las operaciones exactas realizadas por esos componentes pueden variar según el tipo de información que se comprime.

Si el fotograma (505) actual es un fotograma predicho, un estimador (510) de movimiento estima el movimiento de bloques, sub-bloques u otros conjuntos de valores de píxeles del fotograma (505) actual con respecto a uno o más fotogramas de referencia (imágenes de referencia). El almacenamiento (520) de fotogramas almacena en memoria intermedia uno o más fotogramas (525) previos reconstruidos para su uso como fotogramas de referencia (imágenes de referencia). Cuando se utilizan múltiples imágenes de referencia, las múltiples imágenes de referencia pueden ser de diferentes direcciones temporales o de la misma dirección temporal. Las múltiples imágenes de referencia se pueden representar en una o más RPL, que se direccionan con índices de referencia. El estimador (510) de movimiento genera como información lateral información (515) de movimiento tal como información de vector de movimiento diferencial, índices de referencia e información de modificación de RPL. Durante la codificación, el codificador (500) construye la(s) RPL para que se agreguen nuevas imágenes de referencia cuando sea apropiado, las imágenes de referencia más antiguas que ya no se utilizan para la compensación de movimiento se eliminan cuando corresponde, y las imágenes de referencia se reordenan en la(s) RPL cuando sea apropiado.

En algunas implementaciones, al codificar un fotograma actual, el codificador (500) determina un RPS que incluye fotogramas de referencia en el almacenamiento (520) de fotogramas. El codificador (500) determina normalmente el RPS para el primer segmento del fotograma. Segmento a segmento, el codificador (500) crea una o más RPL para codificar un segmento dado del fotograma actual. Para crear una RPL, el codificador (500) puede aplicar reglas sobre la selección de los fotogramas de referencia disponibles del RPS, en cuyo caso la información de modificación de la RPL no se señala explícitamente en los datos (595) codificados. O, para crear la RPL, el codificador (500) puede seleccionar fotogramas de referencia específicos disponibles del RPS, donde los fotogramas de referencia que se seleccionan se indicarán en la información de modificación de RPL que se señala en los datos (595) codificados. En comparación con una RPL que se construiría mediante reglas del enfoque implícito, la información de modificación de la RPL puede especificar una RPL de reemplazo como una lista de imágenes de referencia en el RPS. Alternativamente, la información de modificación de RPL puede, de una manera más detallada, especificar la eliminación de uno o más fotogramas de referencia, la adición de uno o más fotogramas de referencia y/o el reordenamiento de los fotogramas de referencia en la RPL construido implícitamente por reglas.

El compensador (530) de movimiento aplica vectores de movimiento reconstruidos a los fotogramas (525) de referencia reconstruidos cuando se forma un fotograma (535) actual compensado por movimiento. La diferencia (si existe) entre un sub-bloque, bloque, etc. del fotograma (535) actual compensado por movimiento y la parte correspondiente del fotograma (505) actual original es la predicción residual (545) para el sub-bloque, bloque, etc. Durante la reconstrucción posterior del fotograma actual, los residuos de predicción reconstruidos se añaden al fotograma (535) actual compensado por movimiento para obtener un fotograma reconstruido que está más cerca del fotograma (505) actual original. Sin embargo, en la compresión con pérdida, todavía se pierde parte de la información del fotograma (505) actual original. La ruta intra puede incluir un módulo de predicción intra (no mostrado) que predice espacialmente valores de píxeles de un bloque o sub-bloque actual a partir de valores de píxeles vecinos previamente reconstruidos.

Un transformador (560) de frecuencia convierte la información de video de dominio espacial en datos del dominio de frecuencia (es decir, espectrales, de transformación). Para fotogramas de video basados en bloques, el transformador (560) de frecuencia aplica una transformada de coseno discreta, una aproximación entera de la misma u otro tipo de transformada de bloque directo a bloques o sub-bloques de datos de valor de píxel o datos residuales de predicción, produciendo bloques/sub-bloques de coeficientes de transformada de frecuencia. A continuación, un cuantificador (570) cuantifica los coeficientes de transformación. Por ejemplo, el cuantificador (570) aplica una cuantificación escalar no uniforme a los datos del dominio de la frecuencia con un tamaño de paso que varía fotograma por fotograma, segmento por segmento, bloque por bloque u otro.

Cuando se necesita una versión reconstruida del fotograma actual para la estimación/compensación de movimiento subsiguiente, un cuantificador (576) inverso realiza una cuantificación inversa en los datos de coeficientes de frecuencia cuantificados. Un transformador (566) de frecuencia inversa realiza una transformada de frecuencia inversa, al producir bloques/sub-bloques de residuos de predicción reconstruidos o valores de píxeles. Para un fotograma predicho, el codificador (500) combina residuos (545) de predicción reconstruidos con predicciones (535) compensadas por movimiento para formar el fotograma (505) reconstruido, que puede usarse como imagen de referencia. (Aunque no se muestra en la Figura 5, en la ruta intra, el codificador (500) puede combinar los residuales de predicción con predicciones espaciales de la predicción intra para reconstruir un fotograma que se utiliza como imagen de referencia). El almacenamiento (520) de fotogramas almacena en la memoria intermedia el fotograma actual reconstruido para usar como imagen de referencia en la predicción con compensación de movimiento subsiguiente.

Un bucle de compensación de movimiento en el codificador (500) incluye un filtro (510) de desbloqueo en bucle adaptable antes o después del almacenamiento (520) de fotogramas. El decodificador (500) aplica filtrado en bucle a los fotogramas reconstruidos para suavizar adaptativamente las discontinuidades a través de los límites de los fotogramas.

El codificador (580) de entropía comprime la salida del cuantificador (570) así como la información (515) de movimiento y cierta información lateral (por ejemplo, valores de QP, índices de referencia, información de modificación de RPL). El codificador (580) de entropía proporciona datos (595) codificados a la memoria intermedia (590), que multiplexa los

datos codificados en un flujo de bits de salida. Los datos (595) codificados pueden incluir elementos de sintaxis que indican información de modificación de RPL. La Sección VII describe ejemplos de tales elementos de sintaxis.

Un controlador (no mostrado) recibe entradas de varios módulos del codificador. El controlador evalúa los resultados intermedios durante la codificación, por ejemplo, mediante el establecimiento de valores de QP y mediante la realización de un análisis de distorsión de tasa. El controlador trabaja con otros módulos para establecer y cambiar los parámetros de codificación durante la codificación. En particular, al decidir si modificar (por ejemplo, reemplazar, ajustar) la o las RPL, el controlador puede controlar qué imágenes de referencia se agregan a la o las RPL, controlar qué imágenes se eliminan de la(s) RPL y reordenar imágenes de referencia en la(s) RPL para un direccionamiento más eficiente con índices de referencia. El controlador puede decidir eliminar las imágenes de referencia del RPS (y, por lo tanto, las RPL), por ejemplo, mediante la eliminación de todas las imágenes de referencia después de un cambio de escena, mediante la eliminación de todas las imágenes de referencia después de codificar un tipo especial de imagen como una imagen IDR, mediante la eliminación de la imagen de referencia dada después de la utilización de la imagen de referencia para la compensación de movimiento cae por debajo de una cantidad umbral y/o la eliminación de imágenes de referencia según otros criterios. El controlador puede decidir agregar imágenes de referencia al RPS, por ejemplo, mediante la agregación de imágenes según el tipo de imagen/tipos de segmento en las imágenes, la capa temporal para las imágenes y/u otros criterios. Para una RPL, el controlador puede evaluar los resultados de la compensación de movimiento para los que una RPL no se modifica según los elementos de sintaxis señalados explícitamente en el flujo de bits, y también evaluar los resultados de la compensación de movimiento para los cuales la RPL se modifica según los elementos de sintaxis señalados explícitamente en el flujo de bits (o resultados de múltiples formas diferentes de modificar la RPL). El controlador puede evaluar los resultados en términos de tasa de bits y/o calidad. El controlador puede seleccionar la RPL construida implícitamente por reglas (sin información de modificación de RPL) o seleccionar una RPL que haya sido modificada (como se especifica con la información de modificación de RPL). Modificar (por ejemplo, reemplazar, ajustar) una RPL, en comparación con la RPL construida implícitamente, el controlador puede (a) reordenar imágenes de referencia para un direccionamiento más eficiente con índices de referencia, (b) eliminar imágenes de referencia basándose al menos en parte en la frecuencia de uso durante la codificación y/o (c) añadir imágenes de referencia basadas al menos en parte en la frecuencia de uso durante la codificación. Por ejemplo, el controlador puede decidir eliminar una imagen de referencia dada de la RPL después de que la utilización de la imagen de referencia para la compensación de movimiento caiga por debajo de una cantidad umbral y/o según otros criterios. O el controlador puede decidir agregar una imagen de referencia dada a la RPL si la utilización de la imagen de referencia para la compensación de movimiento está por encima de una cantidad umbral y/o según otros criterios. O el controlador puede decidir cómo reordenar las imágenes de referencia en la RPL basándose en la frecuencia de utilización de las respectivas imágenes de referencia y/o según otros criterios. El controlador puede construir la(s) RPL imagen por imagen, segmento por segmento u otro.

Dependiendo de la implementación y el tipo de compresión deseada, los módulos del codificador pueden agregarse, omitirse, dividirse en múltiples módulos, combinarse con otros módulos y/o reemplazarse con módulos similares. En realizaciones alternativas, codificadores con diferentes módulos y/u otras configuraciones de módulos realizan una o más de las técnicas descritas. Las realizaciones específicas de codificadores normalmente usan una variación o versión complementada del codificador (500). Las relaciones mostradas entre módulos dentro del codificador (500) indican flujos generales de información en el codificador; otras relaciones no se muestran en aras de la simplicidad.

#### 40 VI. Decodificadores de video ejemplares.

La Figura 6 es un diagrama de bloques de un decodificador (600) generalizado junto con el cual se pueden implementar varias realizaciones descritas. El decodificador (600) recibe datos (695) codificados para un fotograma comprimido o secuencia de fotogramas y produce una salida que incluye un fotograma (605) reconstruido, que puede usarse como imagen de referencia. En aras de la presentación, la Figura 6 muestra una "ruta intra" a través del decodificador (600) para la decodificación intra-fotograma y una "ruta entre" para la decodificación entre fotogramas. Muchos de los componentes del decodificador (600) se utilizan tanto para la decodificación intra fotograma como para la decodificación entre fotogramas. Las operaciones exactas realizadas por esos componentes pueden variar según el tipo de información que se descomprime.

Una memoria intermedia (690) recibe datos (695) codificados para un fotograma comprimido y hace que los datos codificados recibidos estén disponibles para el analizador sintáctico/decodificador (680) de entropía. Los datos (695) codificados pueden incluir elementos de sintaxis que indican información de modificación de RPL. La Sección VII describe ejemplos de tales elementos de sintaxis. El analizador/decodificador (680) de entropía decodifica en entropía datos cuantificados codificados en entropía así como información lateral codificada en entropía (incluidos índices de referencia, información de modificación de RPL, etc.), mediante la aplicación normalmente de la codificación inversa de entropía realizada en el codificador.

Durante la decodificación, el decodificador (600) construye la(s) RPL para que se agreguen nuevas imágenes de referencia cuando sea apropiado, las imágenes de referencia más antiguas que ya no se usan para la compensación de movimiento se eliminan cuando es apropiado, y las imágenes de referencia se reordenan cuando es apropiado. El decodificador (600) puede construir la(s) RPL basándose en la información disponible sobre la(s) RPL (por ejemplo, imágenes de referencia disponibles en el RPS), modificaciones según reglas y/o según modificaciones señaladas como parte de los datos (695) codificados. En algunas implementaciones, por ejemplo, al decodificar un fotograma

actual, el decodificador (600) determina un RPS que incluye fotogramas de referencia en el almacenamiento (620) de fotogramas. El decodificador (600) determina típicamente el RPS para el primer segmento del fotograma. Segmento a segmento, el decodificador (600) crea una o más RPL para decodificar un segmento dado del fotograma actual. Para crear una RPL, en algunos casos (como se indica en los datos (695) codificados), el decodificador (600) aplica reglas sobre la selección de los fotogramas de referencia disponibles del RPS, en cuyo caso la información de modificación de la RPL no se analiza a partir de los datos (695) codificados. En otros casos, para crear la RPL, el decodificador (600) selecciona fotogramas de referencia específicos disponibles del RPS, donde los fotogramas de referencia que se seleccionan se indican en la información de modificación de RPL que se analiza a partir de los datos (695) codificados. La información de modificación de RPL puede especificar una RPL de reemplazo como una lista de imágenes de referencia en el RPS. Alternativamente, la información de modificación de RPL puede, de una manera más detallada, especificar la eliminación de uno o más fotogramas de referencia, la adición de uno o más fotogramas de referencia y/o el reordenamiento de los fotogramas de referencia en la RPL construida implícitamente por reglas.

Un compensador (630) de movimiento aplica información (615) de movimiento a una o más imágenes (625) de referencia para formar predicciones (635) de movimiento compensado de sub-bloques y/o bloques del fotograma (605) que se está reconstruyendo. El almacenamiento (620) de fotogramas almacena uno o más fotogramas previamente reconstruidos para su uso como imágenes de referencia.

La ruta intra puede incluir un módulo de predicción intra (no mostrado) que predice espacialmente valores de píxeles de un bloque o sub-bloque actual a partir de valores de píxeles vecinos previamente reconstruidos. En ruta entre, el decodificador (600) reconstruye los residuos de predicción. Un cuantificador (670) inverso cuantifica los datos decodificados por entropía. Un transformador (660) de frecuencia inversa convierte los datos de dominio de frecuencia reconstruidos en información de dominio espacial. Por ejemplo, el transformador (660) de frecuencia inversa aplica una transformación de bloque inverso a los coeficientes de transformación de frecuencia, produciendo datos de valor de píxel o datos residuales de predicción. La transformada de frecuencia inversa puede ser una transformada de coseno discreta inversa, una aproximación entera de la misma u otro tipo de transformada de frecuencia inversa.

Para un fotograma predicho, el decodificador (600) combina residuos (645) de predicción reconstruidos con predicciones (635) compensadas por movimiento para formar el fotograma (605) reconstruido, que puede usarse como imagen de referencia. (Aunque no se muestra en la Figura 6, en la ruta intra, el decodificador (600) puede combinar los residuales de predicción con predicciones espaciales de la predicción intra para reconstruir un fotograma, que puede usarse como una imagen de referencia). Un bucle de compensación de movimiento en el decodificador (600) incluye un filtro (610) de desbloqueo en bucle adaptativo antes o después del almacenamiento (620) de fotogramas. El decodificador (600) aplica filtrado en bucle a los fotogramas reconstruidas para suavizar adaptativamente las discontinuidades a través de los límites de los fotogramas.

En la Figura 6, el decodificador (600) también incluye un filtro (608) de desbloqueo de posprocesamiento. El filtro (608) de desbloqueo de posprocesamiento suaviza opcionalmente las discontinuidades en los fotogramas reconstruidos. También se pueden aplicar otros filtros (como el filtrado de anillo) como parte del filtrado de posprocesamiento.

Dependiendo de la implementación y el tipo de descompresión deseada, los módulos del decodificador pueden agregar, omitirse, dividirse en múltiples módulos, combinarse con otros módulos y/o reemplazarse con módulos similares. En realizaciones alternativas, los decodificadores con diferentes módulos y/u otras configuraciones de módulos realizan una o más de las técnicas descritas. Las realizaciones específicas de los decodificadores normalmente usan una variación o una versión complementada del decodificador (600). Las relaciones mostradas entre módulos dentro del decodificador (600) indican flujos generales de información en el decodificador; otras relaciones no se muestran en aras de la simplicidad.

## VII. Señalización de la información de modificación de la lista de imágenes de referencia.

Esta sección presenta varias innovaciones en el área de señalización de información de modificación de RPL. En algunas situaciones, estas innovaciones dan como resultado una señalización más eficiente de los elementos de sintaxis para la información de modificación de RPL.

### A. Imágenes de referencia y RPL.

Una imagen de referencia es, en general, una imagen que contiene muestras que pueden usarse para la predicción entre imágenes en el proceso de decodificación de otras imágenes, que normalmente siguen a la imagen de referencia en el orden de decodificación. Pueden estar disponibles varias imágenes de referencia en un momento dado para su uso en la predicción con compensación de movimiento.

En general, una lista de imágenes de referencia ("RPL") es una lista de imágenes de referencia utilizadas para la predicción con compensación de movimiento. Las imágenes de referencia en la RPL se tratan con índices de referencia. Un índice de referencia identifica una imagen de referencia en la RPL. Durante la codificación y decodificación, cuando se construye una RPL, las imágenes de referencia en la RPL pueden cambiar de vez en cuando para agregar imágenes recién decodificadas, eliminar imágenes más antiguas que ya no se utilizan como imágenes de referencia y/o reordenar imágenes de referencia dentro de la RPL para hacer que la señalización de los índices de referencia más utilizados sea más eficiente. Un codificador y un decodificador pueden seguir las mismas reglas para

construir, modificar, etc. su(s) RPL. Además de tales reglas (o en lugar de tales reglas), un codificador puede enviar información a un decodificador que indica cómo el decodificador debe construir, modificar, etc. su(s) RPL para que coincidan con la(s) RPL utilizadas por el codificador. Normalmente, una RPL se construye durante la codificación y decodificación en función de la información disponible sobre la RPL (por ejemplo, imágenes disponibles en el RPS), modificaciones según reglas y/o modificaciones señaladas en el flujo de bits.

En algunas implementaciones, para una imagen actual, un codificador o decodificador determina un conjunto de imágenes de referencia ("RPS") que incluye imágenes de referencia en un área de almacenamiento de fotogramas decodificados como una memoria intermedia de imágenes decodificadas ("DPB"). El RPS es una descripción de las imágenes de referencia utilizadas en el proceso de decodificación de las imágenes codificadas actuales y futuras. Las imágenes de referencia incluidas en el RPS se enumeran explícitamente en el flujo de bits.

El codificador o decodificador determina el RPS una vez por imagen. Por ejemplo, el decodificador determina el RPS después de decodificar una cabecera de segmento para un segmento de la imagen, mediante el uso de elementos de sintaxis señalados en la cabecera de segmento. Las imágenes de referencia se identifican con valores de recuento de orden de imágenes ("POC"), partes de las mismas y/u otra información señalizada en el flujo de bits. El codificador o decodificador determina grupos de imágenes de referencia a corto plazo e imágenes de referencia a largo plazo que pueden usarse en la predicción entre imágenes de la imagen actual (y que pueden usarse en la predicción entre imágenes de una o más de las imágenes siguientes a la imagen actual en orden de decodificación). (El codificador o decodificador también determina grupos de imágenes de referencia que pueden usarse en la predicción entre imágenes de una o más de las imágenes que siguen a la imagen actual en el orden de decodificación, pero no se usan para la imagen actual). Las imágenes de referencia son los RPS de la imagen actual.

Para un segmento determinado de la imagen actual, el codificador o decodificador crea una o más RPL. El codificador o decodificador crea una versión temporal de una RPL (por ejemplo, RPL 0 o RPL 1) al combinar los grupos de imágenes de referencia a corto plazo e imágenes de referencia a largo plazo que pueden usarse en la predicción entre imágenes de la imagen actual. Para construir la RPL según las reglas de un enfoque "implícito", el codificador o decodificador puede usar las imágenes de referencia en la versión temporal de la RPL, o usar solo algunas de las imágenes de referencia en la versión temporal de la RPL (por ejemplo, las primeras x imágenes en la versión temporal de la RPL). Para el enfoque "implícito", la información de modificación de RPL no se señalizará en el flujo de bits y no se analizará a partir del flujo de bits. En un enfoque "explícito", para construir la RPL, el codificador o decodificador usa información de modificación de RPL señalada/analizada desde el flujo de bits para seleccionar imágenes de referencia específicas de la versión temporal de la RPL. En comparación con la RPL que se construiría mediante las reglas del enfoque "implícito", la información de modificación de la RPL puede especificar la eliminación de una o más imágenes de referencia, la adición de una o más imágenes de referencia y/o el reordenamiento de las imágenes de referencia en la RPL.

Alternativamente, un codificador o decodificador utiliza otro enfoque para crear una RPL a partir de imágenes de referencia.

#### B. Señalización condicional de banderas de modificación de RPL.

Según un aspecto de las innovaciones descritas en este documento, un codificador señaliza condicionalmente una bandera que indica si una RPL se modifica según elementos de sintaxis señalados explícitamente en el flujo de bits. Un decodificador correspondiente analiza condicionalmente dicha bandera.

En algunas implementaciones ejemplares, la bandera es bandera\_modificación\_lista\_imag\_ref\_I0 o bandera\_modificación\_lista\_imag\_ref\_I1 (generalmente, la bandera es bandera\_modificación\_lista\_imag\_ref\_IX, donde X puede ser 0 o 1). Si el valor de la bandera bandera\_modificación\_lista\_imag\_ref\_IX es igual a 1, la RPL X se especifica explícitamente como una lista de valores entrada\_lista\_IX[i] (nuevamente, siendo X 0 o 1). Si el valor de la bandera bandera\_modificación\_lista\_imag\_ref\_IX es igual a 0, la RPL X se determina implícitamente. Cuando bandera\_modificación\_lista\_imag\_ref\_IX no está presente, se infiere que es igual a 0.

La Figura 7a muestra un ejemplo de sintaxis (700) para una estructura de sintaxis modificación\_listas\_imag\_ref() en implementaciones ejemplares. La estructura se puede señalar como parte de una cabecera de segmento. En la sintaxis (700) ejemplar, bandera\_modificación\_lista\_imag\_ref\_IX solo se envía cuando NumPocTotalAct es mayor que 1. NumPocTotalAct es una variable que indica un número total de imágenes de referencia aplicables para la codificación o decodificación actual. En implementaciones ejemplares de codificación o decodificación, cuando la variable NumPocTotalAct se deriva para un segmento de una imagen actual, la variable indica el recuento de imágenes de referencia a corto plazo e imágenes de referencia a largo plazo utilizadas como imágenes de referencia para codificar o decodificar la imagen.

Como se muestra en la Figura 7a, la señalización condicional de

bandera\_modificación\_lista\_imag\_ref\_IX depende del valor de la variable NumPocTotalAct. Cuando NumPocTotalAct es menor o igual a 1, no hay posibilidad de modificación de la RPL y, por lo tanto, no es necesario enviar la bandera\_modificación\_lista\_imag\_ref\_IX. Esta señalización condicional puede guardar una o dos banderas por cada segmento, cuando se cumple la condición. La modificación en la Figura 7a incluye la condición "si (NumPocTotalAct >

1)" para si la bandera bandera\_modificación\_lista\_imag\_ref\_IX está señalizada. La condición se puede verificar para la lista 0 (para un segmento P o un segmento B) y/o para la lista 1 (para un segmento B).

De manera alternativa, la señalización y análisis sintáctico de una estructura de modificación de RPL que incluye una o más banderas de modificación de RPL (por ejemplo, una estructura modificación\_listas\_imag\_ref()) se puede

5 controlar mediante la evaluación de una condición como parte del procesamiento de la cabecera del segmento o de otro modo. La Figura 7b ilustra un enfoque para la señalización condicional y el análisis sintáctico de las banderas bandera\_modificación\_lista\_imag\_ref\_I0 y bandera\_modificación\_lista\_imag\_ref\_I1 basado en esta condición. Específicamente, la Figura 7b muestra una sintaxis (750) ejemplar para una estructura de sintaxis de cabecera de segmento que puede incluir una estructura de sintaxis modificación\_listas\_imag\_ref(), que se representa en la sintaxis 10 (760) de la Figura 7c. Para la sintaxis (750) ejemplar de la cabecera del segmento, la bandera bandera\_presente\_modificación\_listas se señala en un conjunto de parámetros de imagen que se aplica al segmento. Cuando bandera\_presente\_modificación\_listas es igual a 0, la estructura modificación\_listas\_imag\_ref() no está 15 presente en la cabecera del segmento. Cuando bandera\_presente\_modificación\_listas es igual a 1, la estructura modificación\_listas\_imag\_ref() puede estar presente en la cabecera del segmento, dependiendo del valor de la variable NumPocTotalAct. Si la variable NumPocTotalAct es mayor que 1, entonces se señala la estructura modificación\_listas\_imag\_ref(), como se muestra en la sintaxis (760) de la Figura 7c. De lo contrario (la variable NumPocTotalAct no es mayor que 1), la estructura modificación\_listas\_imag\_ref() no se señala y se infieren los valores de las entradas de la lista.

20 En las Figuras 7a-7c, 8 y 9, el término "u(n)" representa un número entero sin signo que utiliza n bits. Cuando n es "v" (como en "u(v)"), el número de bits varía de una manera que depende del valor de otros elementos de sintaxis. El proceso de análisis sintáctico para u(n) se puede especificar mediante el valor de retorno de una función que lee n bits como una representación binaria de un entero sin signo, con el bit más significativo escrito primero.

#### C. Señalización de elementos de sintaxis para entradas de lista.

25 Según otro aspecto de las innovaciones aquí descritas, un codificador señala condicionalmente elementos de sintaxis para entradas de lista que indican cómo modificar una RPL. Un decodificador correspondiente analiza condicionalmente dichos elementos de sintaxis.

30 En algunas implementaciones ejemplares, los elementos de sintaxis son para entrada\_list\_I0[i] elementos de sintaxis para RPL 0 o entrada\_list\_I1[i] elementos de sintaxis para RPL 1 (generalmente, el elemento de sintaxis es entrada\_list\_IX, donde X puede ser 0 o 1). La Figura 8 muestra una sintaxis (800) para una estructura de sintaxis modificación\_listas\_imag\_ref(), que se puede señalar como parte de una cabecera de segmento. En la sintaxis 35 ejemplar (800), el elemento de sintaxis entrada\_list\_IX[0] se señala condicionalmente en el flujo de bits. En particular, cuando NumPocTotalAct es igual a 2 y num\_ref\_idx\_IX\_activo\_menos1 es igual a 0, el elemento de sintaxis entrada\_list\_IX[0] no se señala en el flujo de bits. La variable num\_ref\_idx\_IX\_activo\_menos1 indica el índice de referencia máximo para la RPL X que puede usarse para decodificar un segmento. La variable num\_ref\_idx\_IX\_activo\_menos1 puede tener un valor predeterminado (por ejemplo, un valor de 0 ... 15, como se especifica en el conjunto de parámetros de imagen aplicable), o

40 num\_ref\_idx\_IX\_activo\_menos1 puede tener un valor señalado en una cabecera de segmento para el segmento actual.

45 Como se muestra en la Figura 8, incluso cuando bandera\_modificación\_lista\_imag\_ref\_IX indica que la información de modificación de RPL está señalizada en el flujo de bits, la señalización de entrada\_list\_IX[0] depende de NumPocTotalAct y num\_ref\_idx\_IX\_activo\_menos1. Cuando NumPocTotalAct es igual a 2 y num\_ref\_idx\_IX\_activo\_menos1 es igual a 1, el valor de entrada\_list\_IX[0] puede inferirse basándose en bandera\_modificación\_lista\_imag\_ref\_IX, ya que solo hay dos opciones posibles (valor predeterminado de 0 o el valor no predeterminado de 1).

50 Por tanto, la Figura 8 incluye una condición para si los elementos de sintaxis para las entradas de la lista están señalizados. Para RPL 0, la condición es "si (bandera\_modificación\_lista\_imag\_ref\_I0 && (NumPocTotalAct == 2 && num\_ref\_idx\_I0\_activo\_menos1 == 0))." Para RPL 1, la condición es "si(bandera\_modificación\_lista\_imag\_ref\_I1 && (NumPocTotalAct == 2 && num\_ref\_idx\_I1\_activo\_menos1 == 0))".

55 En el ejemplo de la Figura 8, entrada\_list\_I0[i] especifica el índice de la imagen de referencia en RefImgListaTemp0 (una versión temporal de RPL) que se colocará en la posición actual de RPL 0. La longitud del elemento de sintaxis entrada\_list\_I0[i] es Ceil (Log2 (NumPocTotalAct)) bits. El valor de entrada\_list\_I0[i] está en el rango de 0 a NumPocTotalAct - 1, inclusive. Si NumPocTotalAct es igual a 2 y num\_ref\_idx\_I0\_activo\_menos1 es igual a 0, se infiere que el elemento de sintaxis entrada\_list\_I0[0] es igual a bandera\_modificación\_lista\_imag\_ref\_I0. De lo contrario, cuando el elemento de sintaxis entrada\_list\_I0[i] no está presente, se infiere que es igual a 0.

55 En el ejemplo de la Figura 8, entrada\_list\_I1[i] especifica el índice de la imagen de referencia en RefImgListaTemp1 (una versión temporal de RPL) que se colocará en la posición actual de RPL 1. La longitud del elemento de sintaxis entrada\_list\_I1[i] es Ceil (Log2 (NumPocTotalAct)) bits. El valor de entrada\_list\_I1[i] está en el rango de 0 a NumPocTotalAct - 1, inclusive. Si NumPocTotalAct es igual a 2 y num\_ref\_idx\_I1\_activo\_menos1 es igual a 0, se infiere

que el elemento de sintaxis `entrada_listा_I1[0]` es igual a `bandera_modificación_listा_imag_ref_I1`. De lo contrario, cuando la lista de elementos de sintaxis `entrada_listा_I1[i]` no está presente, se infiere que es igual a 0.

La Figura 9 muestra otra sintaxis (900) ejemplar para una estructura de sintaxis `modificación_listा_imag_ref()`, que puede indicarse como parte de una cabecera de segmento. En la sintaxis (900) ejemplar, el elemento de sintaxis `entrada_listा_IX[0]` se señala condicionalmente en el flujo de bits. Sin embargo, en comparación con la sintaxis ejemplar de la Figura 8, la condición que se marca es diferente. Además, la señalización de los elementos de sintaxis para `entrada_listा_IX[]` puede ajustarse dependiendo de si se utiliza la predicción ponderada.

Según la Figura 9, el hecho de que la predicción ponderada esté habilitada o deshabilitada afecta la forma en que los elementos de sintaxis para las entradas de la lista se señalizan en el flujo de bits. Para los segmentos P con `bandera_pred_ponderada` igual a 0 o para los segmentos B con `bandera_bipred_ponderada` igual a 0, la predicción ponderada está deshabilitada. Según la sintaxis (900) ejemplar de la Figura 9, cuando la predicción ponderada está deshabilitada, `entrada_listा_IX[0]` y `entrada_listा_IX[1]` no se envían cuando `NumPocTotalAct` es igual a 2 y `num_ref_idx_IX_activo_menos1` es igual a 1. En tal caso, `entrada_listा_IX[0]` y `entrada_listा_IX[1]` se infieren como 1 y 0, respectivamente, ya que la modificación de RPL no habría sido necesaria para la única otra posibilidad (es decir, `entrada_listा_IX[0]` y `entrada_listा_IX[1]` siendo iguales a 0 y 1, respectivamente).

Por tanto, la Figura 9 incluye una condición para si los elementos de sintaxis para las entradas de la lista están señalizados. Para RPL 0, la condición es "si(`bandera_modificación_listा_imag_ref_I0` && (`NumPocTotalAct` == 2 && `num_ref_idx_I0_activo_menos1` == 0) && (`NumPocTotalAct` == 2 && `num_ref_idx_I0_activo_menos1` == 1 && (`bandera_bipred_ponderada` != 1 && `tipo_segmento` == P || (`bandera_bipred_ponderada` != 1 && `tipo_segmento` == B)))". Para RPL 1, la condición es "si (`bandera_modificación_listा_imag_ref_I1` && (`NumPocTotalAct` == 2 && `num_ref_idx_I1_activo_menos1` == 0) && (`NumPocTotalAct` == 2 && `num_ref_idx_I1_activo_menos1` == 1 && `bandera_bipred_ponderada` != 1))".

Además, incluso en los casos en los que `NumPocTotalAct` no es igual a 2 o `num_ref_idx_IX_activo_menos1` no es igual a 1, cuando la predicción ponderada está deshabilitada (para segmentos P, `bandera_pred_ponderada` igual a 0; para segmentos B, `bandera_bipred_ponderada` igual a 0), la longitud del elemento de sintaxis `entrada_listा_IX[i]` está limitado a `Ceil(Log2 (NumPocTotalAct-i))` bits. En este caso, solo es útil colocar cada imagen de referencia una vez en la lista y, por lo tanto, el número de posibilidades útiles disminuye a medida que aumenta el índice i.

En el ejemplo de la Figura 9, `entrada_listा_I0[i]` especifica el índice de la imagen de referencia en `ReflmagListaTemp0` (una versión temporal de RPL) que se colocará en la posición actual de RPL 0. Cuando la predicción ponderada está deshabilitada (para segmentos P, `bandera_pred_ponderada` igual a 0; para segmentos B, `bandera_bipred_ponderada` igual a 0), la longitud del elemento de sintaxis `entrada_listा_I0[i]` es `Ceil(Log2 (NumPocTotalAct-i))` bits. De lo contrario, la longitud del elemento de sintaxis `entrada_listा_I0[i]` es `Ceil(Log2 (NumPocTotalAct))` bits. Si `NumPocTotalAct` es igual a 2 y `num_ref_idx_I0_activo_menos1` es igual a 0, se infiere que el elemento de sintaxis `entrada_listा_I0[0]` es igual a `bandera_modificación_listा_imag_ref_I0` (como en el ejemplo de la Figura 8). De lo contrario, si `NumPocTotalAct` es igual a 2, `num_ref_idx_I0_activo_menos1` es igual a 1 y la predicción ponderada está deshabilitada (cuando `bandera_pred_ponderada` es igual a 0 y el segmento actual es un segmento P, o `bandera_bipred_ponderada` es igual a 0 y el segmento actual es un segmento B), se infiere que los elementos de sintaxis `entrada_listा_I0[0]` y `entrada_listा_I0[1]` son iguales a 1 y 0 respectivamente. De lo contrario, cuando el elemento de sintaxis `entrada_listा_I0[i]` no está presente, se infiere que es igual a 0.

Si la predicción ponderada está deshabilitada (cuando `bandera_pred_ponderada` es igual a 0 y el segmento actual es un segmento P, o cuando el segmento `bandera_bipred_ponderada` es igual a 0 y el segmento actual es un segmento B), el valor de `entrada_listा_I0[i]` está en el rango de 0 a `NumPocTotalAct - (i + 1)`, inclusive, y la lista `ReflmagListaTemp0` se acorta al eliminar cada entrada `entrada_listा_I0[i]` de la lista `ReflmagListaTemp0` después de analizar el valor de la entrada. De lo contrario, el valor de `entrada_listा_I0[i]` está en el rango de 0 a `NumPocTotalAct-1`, inclusive.

En el ejemplo de la Figura 9, `entrada_listा_I1[i]` especifica el índice de la imagen de referencia en `ReflmagListaTemp1` (una versión temporal de RPL) que se colocará en la posición actual de RPL 1. Si la predicción ponderada está deshabilitada (`bandera_bipred_ponderada` es igual a 0, dado que solo un segmento B usa la lista 1), la longitud del elemento de sintaxis `entrada_listा_I1[i]` es `Ceil(Log2 (NumPocTotalAct-i))` bits. De lo contrario, la longitud del elemento de sintaxis `entrada_listा_I1[i]` de la lista es `Ceil(Log2 (NumPocTotalAct))` bits. Si `NumPocTotalAct` es igual a 2 y `num_ref_idx_I1_activo_menos1` es igual a 0, se infiere que el elemento de sintaxis `entrada_listा_I1[0]` es igual a `bandera_modificación_listा_imag_ref_I1` (como en el ejemplo de la Figura 8). De lo contrario, si `NumPocTotalAct` es igual a 2, `num_ref_idx_I1_activo_menos1` es igual a 1 y la predicción ponderada está deshabilitada (`bandera_bipred_ponderada` es igual a 0 - el segmento actual es un segmento B), se infieren los elementos de sintaxis `entrada_listा_I1[0]` y `entrada_listा_I1[1]` para ser igual a 1 y 0 respectivamente. De lo contrario, cuando el elemento de sintaxis `entrada_listा_I1[i]` no está presente, se infiere que es igual a 0.

Si la predicción ponderada está deshabilitada (`bandera_bipred_ponderada` es igual a 0; el segmento actual es un segmento B), el valor de `entrada_listा_I1[i]` está en el rango de 0 a `NumPocTotalAct - (i + 1)`, inclusive, y la lista `ReflmagListaTemp1` se acorta mediante la eliminación de cada entrada `entrada_listा_I1[i]` de la lista

ReflMagListaTemp1 después de analizar el valor de la entrada. De lo contrario, el valor de entrada\_listा\_l0 [i] está en el rango de 0 a NumPocTotalAct - 1, inclusive.

D. Técnicas generalizadas para señalización y análisis condicional de banderas de modificación RPL.

5 La Figura 10 muestra una técnica (1000) generalizada para la señalización condicional de una bandera de modificación de RPL. Un dispositivo informático que implementa un codificador de video, por ejemplo, como se describe con referencia a la Figura 3, puede realizar la técnica (1000).

10 El dispositivo evalúa (1010) una condición. Por ejemplo, la condición depende, al menos en parte, de una variable que indica un número total de imágenes de referencia. En algunas implementaciones ejemplares, la variable es NumPocTotalAct y el codificador verifica si la variable es mayor que 1. Alternativamente, el codificador evalúa otras y/o condiciones adicionales. La condición que se evalúa (1010) puede incluir un solo factor (por ejemplo, valor de la variable que indica un número total de imágenes de referencia), o la condición que se evalúa (1010) puede incluir múltiples factores (por ejemplo, valor de la variable que indica un número total de imágenes de referencia, así como uno o más factores). La condición se puede evaluar (1010) como parte del procesamiento de una estructura de modificación de RPL. O bien, la condición se puede evaluar (1010) como parte del procesamiento de una cabecera de segmento.

15 Dependiendo de los resultados de la evaluación, el dispositivo señaliza (1020) condicionalmente en un flujo de bits una bandera que indica si se modifica una RPL (por ejemplo, se reemplaza, se ajusta) según los elementos de sintaxis señalados explícitamente en el flujo de bits. Por ejemplo, la bandera es una de bandera\_modificación\_listा\_imag\_ref\_l0 o bandera\_modificación\_listा\_imag\_ref\_l1, y se puede señalizar condicionalmente como parte de una estructura de modificación RPL de una cabecera de segmento. O, después de evaluar (1010) la condición, dependiendo de los resultados de la evaluación, la estructura de modificación de RPL (que incluye una o más banderas que indican si una RPL se modifica según elementos de sintaxis señalados explícitamente en el flujo de bits) se señaliza condicionalmente en el flujo de bits.

20 25 El dispositivo puede repetir la técnica (1000) segmento a segmento cuando se señala la estructura de modificación de RPL, o sobre alguna otra base.

20 La Figura 11 muestra una técnica (1100) generalizada para el análisis sintáctico condicional de una bandera de modificación de RPL. Un dispositivo informático que implementa un decodificador de video, por ejemplo, como se describe con referencia a la Figura 4, puede realizar la técnica (1100).

30 35 El decodificador evalúa (1110) una condición. Por ejemplo, la condición depende, al menos en parte, de una variable que indica un número total de imágenes de referencia. En algunas implementaciones ejemplares, la variable es NumPocTotalAct y el decodificador verifica si la variable es mayor que 1. Alternativamente, el decodificador evalúa otras y/o condiciones adicionales. La condición que se evalúa (1110) puede incluir un solo factor (por ejemplo, el valor de la variable que indica un número total de imágenes de referencia), o la condición que se evalúa (1110) puede incluir múltiples factores (por ejemplo, valor de la variable que indica un número total de imágenes de referencia, así como uno o más factores). La condición se puede evaluar (1110) como parte del procesamiento de una estructura de modificación de RPL. O bien, la condición se puede evaluar (1110) como parte del procesamiento de una cabecera de segmento.

40 45 Dependiendo de los resultados de la evaluación, el dispositivo analiza (1120) condicionalmente a partir de un flujo de bits una bandera que indica si una RPL se modifica (por ejemplo, se reemplaza, se ajusta) según elementos de sintaxis señalados explícitamente en el flujo de bits. Por ejemplo, la bandera es una de bandera\_modificación\_listा\_imag\_ref\_l0 o bandera\_modificación\_listा\_imag\_ref\_l1, y se puede señalizar condicionalmente como parte de una estructura de modificación RPL de una cabecera de segmento. O, después de evaluar (1110) la condición, dependiendo de los resultados de la evaluación, la estructura de modificación de RPL (que incluye una o más banderas que indican si una RPL se modifica según elementos de sintaxis señalados explícitamente en el flujo de bits) se analiza condicionalmente del flujo de bits.

45 El dispositivo puede repetir la técnica (1100) segmento a segmento cuando se señala la estructura de modificación de RPL, o sobre alguna otra base.

E. Técnicas generalizadas para la señalización y análisis condicional de entradas de listas.

50 La Figura 12 muestra una técnica (1200) generalizada para la señalización condicional de entradas de lista para la modificación de RPL. Un dispositivo informático que implementa un codificador de video, por ejemplo, como se describe con referencia a la Figura 3, puede realizar la técnica (1200).

55 El dispositivo evalúa (1210) una condición. Por ejemplo, la condición depende al menos en parte de una variable que indica un número total de imágenes de referencia (por ejemplo, NumPocTotalAct en algunas implementaciones ejemplares). O bien, la condición depende, al menos en parte, de una serie de imágenes de referencia activas para la RPL. O bien, la condición depende, al menos en parte, de si la predicción ponderada está deshabilitada. Se puede usar una lógica diferente para verificar si la predicción ponderada está deshabilitada dependiendo de si un segmento

actual es un segmento P o un segmento B y/o dependiendo de qué RPL se esté señalando/analizando. Por ejemplo, la lógica para verificar la condición de una primera RPL (que podría ser utilizada por un segmento P o un segmento B) es diferente a la lógica para verificar la condición de una segunda RPL (que solo puede ser utilizada por un segmento B). Alternativamente, el codificador evalúa otras y/o condiciones adicionales.

5 Dependiendo de los resultados de la evaluación, el dispositivo señaliza (1220) condicionalmente en un flujo de bits uno o más elementos de sintaxis para entradas de lista que indican cómo modificar (por ejemplo, reemplazar, ajustar) una RPL. Por ejemplo, el/los elemento(s) de sintaxis para las entradas de la lista se señala(n) condicionalmente como parte de una estructura de modificación RPL de una cabecera de segmento.

10 En algunas implementaciones exemplares, si (a) el número total de imágenes de referencia es igual a 2 y (b) el número de imágenes de referencia activas para la RPL es igual a 1, entonces el/los elemento(s) de sintaxis para las entradas de la lista están(n) ausentes del flujo de bits, y se infiere un valor para una de las entradas de la lista. En otras implementaciones exemplares, además de esta condición, si (c) el número de imágenes de referencia totales es igual a 2, (d) el número de imágenes de referencia activas para la RPL es igual a 2 y (e) la predicción ponderada está deshabilitada, entonces uno o más elementos de sintaxis para las entradas de la lista están ausentes del flujo de bits y se infieren valores para dos de las entradas de la lista.

15 El dispositivo puede repetir la técnica (1200) segmento a segmento cuando se señala la estructura de modificación de RPL, o sobre alguna otra base.

20 La Figura 13 muestra una técnica (1300) generalizada para el análisis sintáctico condicional de las entradas de la lista para la modificación de RPL. Un dispositivo informático que implementa un decodificador de video, por ejemplo, como se describe con referencia a la Figura 4, puede realizar la técnica (1300).

25 El decodificador evalúa (1310) una condición. Por ejemplo, la condición depende al menos en parte de una variable que indica un número total de imágenes de referencia (por ejemplo, NumPocTotalAct en algunas implementaciones exemplares). O bien, la condición depende, al menos en parte, de una serie de imágenes de referencia activas para la RPL. O bien, la condición depende, al menos en parte, de si la predicción ponderada está deshabilitada. Se puede usar una lógica diferente para verificar si la predicción ponderada está deshabilitada dependiendo de si un segmento actual es un segmento P o un segmento B y/o dependiendo de qué RPL se esté señalando/analizando. Alternativamente, el decodificador evalúa otras y/o condiciones adicionales.

30 Dependiendo de los resultados de la evaluación, el dispositivo analiza (1320) condicionalmente de un flujo de bits uno o más elementos de sintaxis para las entradas de la lista que indican cómo modificar (por ejemplo, reemplazar, ajustar) una RPL. Por ejemplo, el/los elemento(s) de sintaxis para las entradas de la lista se analiza(n) condicionalmente a partir de una estructura de modificación RPL de una cabecera de segmento.

35 En algunas implementaciones exemplares, si (a) el número total de imágenes de referencia es igual a 2 y (b) el número de imágenes de referencia activas para la RPL es igual a 1, entonces los elementos de sintaxis para las entradas de la lista están ausentes del flujo de bits, y se infiere un valor para una de las entradas de la lista. En otras implementaciones exemplares, además de esta condición, si (c) el número de imágenes de referencia totales es igual a 2, (d) el número de imágenes de referencia activas para la RPL es igual a 2 y (e) la predicción ponderada está deshabilitada, entonces uno o más elementos de sintaxis para las entradas de la lista están ausentes del flujo de bits y se infieren valores para dos de las entradas de la lista.

40 El dispositivo puede repetir la técnica (1300) segmento a segmento cuando se señala la estructura de modificación de RPL, o sobre alguna otra base.

#### F. Técnicas generalizadas para ajustar la señalización y el análisis de entradas de listas.

La Figura 14 muestra una técnica (1400) generalizada para ajustar la señalización de las entradas de la lista para la modificación de RPL. Un dispositivo informático que implementa un codificador de video, por ejemplo, como se describe con referencia a la Figura 3, puede realizar la técnica (1400).

45 El dispositivo evalúa (1410) una condición. Por ejemplo, la condición depende, al menos en parte, de si la predicción ponderada está inhabilitada. Se puede usar una lógica diferente para verificar si la predicción ponderada está deshabilitada dependiendo de si un segmento actual es un segmento P o un segmento B y/o dependiendo de qué RPL se esté señalando/analizando. Por ejemplo, la lógica para verificar la condición de una primera RPL (que podría ser utilizada por un segmento P o un segmento B) es diferente a la lógica para verificar la condición de una segunda RPL (que solo puede ser utilizada por un segmento B). Alternativamente, el codificador evalúa otras y/o condiciones adicionales.

55 Dependiendo de los resultados de la evaluación, el dispositivo ajusta (1420) la señalización en un flujo de bits de uno o más elementos de sintaxis para las entradas de la lista que indican cómo modificar (por ejemplo, reemplazar, ajustar) una RPL. En particular, se ajusta la longitud (en bits) de al menos uno del/de los elemento(s) de sintaxis. Por ejemplo, para un índice  $i$  para las entradas de la lista, si la predicción ponderada está deshabilitada, la longitud (en bits) de al menos uno de los elementos de sintaxis disminuye a medida que  $i$  aumenta. En algunas implementaciones exemplares,

si la predicción ponderada está deshabilitada, la longitud de un elemento de sintaxis dado para la entrada de la lista [i] es  $\text{Ceil}(\text{Log2}(\text{NumPocTotalAct}-i))$ . De lo contrario (la predicción ponderada está habilitada), la longitud del elemento de sintaxis dado para la entrada de la lista [i] es  $\text{Ceil}(\text{Log2}(\text{NumPocTotalAct}))$  bits.

5 El dispositivo puede repetir la técnica (1400) segmento a segmento cuando se señala la estructura de modificación de RPL, o sobre alguna otra base.

La Figura 15 muestra una técnica (1500) generalizada para ajustar el análisis sintáctico de las entradas de la lista para la modificación de RPL. Un dispositivo informático que implementa un decodificador de video, por ejemplo, como se describe con referencia a la Figura 4, puede realizar la técnica (1500).

10 El decodificador evalúa (1510) una condición. Por ejemplo, la condición depende, al menos en parte, de si la predicción ponderada está inhabilitada. Se puede usar una lógica diferente para verificar si la predicción ponderada está deshabilitada dependiendo de si un segmento actual es un segmento P o un segmento B y/o dependiendo de qué RPL se esté señalando/analizando. Alternativamente, el decodificador evalúa otras y/o condiciones adicionales.

15 Dependiendo de los resultados de la evaluación, el dispositivo ajusta (1520) el análisis sintáctico de un flujo de bits de uno o más elementos de sintaxis para las entradas de la lista que indican cómo modificar (por ejemplo, reemplazar, ajustar) una RPL. En particular, se ajusta la longitud (en bits) de al menos uno de los elementos de sintaxis. Por ejemplo, para un índice i para las entradas de la lista, si la predicción ponderada está deshabilitada, la longitud (en bits) del al menos uno de los elementos de sintaxis disminuye a medida que aumenta i. En algunas implementaciones exemplares, si la predicción ponderada está deshabilitada, la longitud de un elemento de sintaxis dado para la entrada de la lista [i] es  $\text{Ceil}(\text{Log2}(\text{NumPocTotalAct}-i))$ . De lo contrario (la predicción ponderada está habilitada), la longitud del elemento de sintaxis dado para la entrada de la lista [i] es  $\text{Ceil}(\text{Log2}(\text{NumPocTotalAct}))$  bits.

20 El dispositivo puede repetir la técnica (1500) segmento a segmento cuando se señala la estructura de modificación de RPL, o sobre alguna otra base.

#### G. Alternativas.

25 Las Figuras 7a, 7b, 10 y 11 ilustran la señalización y el análisis sintáctico condicional de una bandera tal como bandera\_modificación\_lista\_imag\_ref\_l0 o bandera\_modificación\_lista\_imag\_ref\_l1 en base a una condición. De esta manera, se controla la señalización de información de modificación de RPL adicional (como elementos de sintaxis para entradas de lista). Como se explica con referencia a la Figura 7a, la señalización y el análisis sintáctico de una bandera de modificación de RPL pueden controlarse mediante la evaluación de la condición como parte de una estructura modificación\_listas\_imag\_ref(). Alternativamente, como se explica con referencia a la Figura 7b, la señalización y análisis sintáctico de la estructura de modificación de RPL (por ejemplo, estructura modificación\_listas\_imag\_ref()) se puede controlar mediante la evaluación de la misma condición como parte del procesamiento de la cabecera del segmento o de otro modo. Por ejemplo, si la variable NumPocTotalAct es mayor que 1, entonces se señala la estructura modificación\_listas\_imag\_ref(). De lo contrario (la variable NumPocTotalAct no es mayor que 1), la estructura modificación\_listas\_imag\_ref() no se señaliza y los valores de las entradas de la lista se infieren como se describe anteriormente con referencia a la Figura 7a. Ampliando la Figura 10, después de que se evalúa la condición, dependiendo de los resultados de la evaluación, la estructura de sintaxis de modificación de RPL se señaliza condicionalmente. Ampliando la Figura 11, después de que se evalúa la condición, dependiendo de los resultados de la evaluación, la estructura de sintaxis de modificación de RPL se analiza condicionalmente.

40 A modo de ilustración, la descripción detallada incluye varios ejemplos con nombres específicos para algunos parámetros y variables. Las innovaciones aquí descritas no se limitan a implementaciones con parámetros o variables que tengan tales nombres. En cambio, las innovaciones descritas en este documento se pueden implementar con varios tipos de parámetros y variables.

## REIVINDICACIONES

1. Un método (1000) realizado por un codificador (340; 500) de video, que comprende:
 

evaluar (1010) si un valor de una variable que indica un número de imágenes de referencia totales es mayor que 1; y

5 solo cuando el valor es mayor que 1, señalizar (1020) condicionalmente en un flujo de bits una estructura de modificación de una lista de imágenes de referencia, RPL, donde la estructura de modificación de RPL incluye una bandera que indica si una RPL se modifica según elementos de sintaxis señalados explícitamente en el flujo de bits como una lista de entradas de la lista, en la que cada una de las entradas de la lista especifica un índice de una imagen de referencia en una versión temporal de la RPL que se colocará en una posición actual de la RPL,

10 en donde si el valor de dicha bandera es igual a 1, la RPL se especifica explícitamente como la lista, y si el valor de dicha bandera es igual a 0, la RPL se determina implícitamente.
2. El método de la reivindicación 1 en donde, durante la codificación, el codificador (340; 500) de video:
 

evalúa los resultados de la compensación de movimiento para los que se modifica la RPL según los elementos de sintaxis señalados explícitamente en el flujo de bits;

15 evalúa los resultados de la compensación de movimiento para los que la RPL no se modifica según los elementos de sintaxis señalados explícitamente en el flujo de bits; y

modifica la RPL para:
 
  - reordenar una o más imágenes de referencia para un direccionamiento más eficiente con índices de referencia,
  - 20 eliminar una o más imágenes de referencia basadas, al menos en parte, en la frecuencia de uso durante la codificación, y/o
  - agregar una o más imágenes de referencia basadas, al menos en parte, en la frecuencia de uso durante la codificación.
3. Un dispositivo informático que implementa un codificador de video, en donde el dispositivo informático está adaptado para realizar un método según una de las reivindicaciones anteriores.
- 25 4. Un método (1100) realizado por un decodificador (450; 600) de video, que comprende:
 

evaluar (1110) si un valor de una variable que indica un número de imágenes de referencia totales es mayor que 1; y

30 solo cuando el valor es mayor que 1, analizar (1120) condicionalmente de un flujo de bits una estructura de modificación de una lista de imágenes de referencia, RPL, donde la estructura de modificación de RPL incluye una bandera que indica si una RPL se modifica según elementos de sintaxis explícitamente señalados en el flujo de bits como una lista de entradas de la lista, en la que cada una de las entradas de la lista especifica un índice de una imagen de referencia en una versión temporal de la RPL que se colocará en una posición actual de la RPL,

35 en donde si el valor de dicha bandera es igual a 1, la RPL se especifica explícitamente como la lista, y si el valor de dicha bandera es igual a 0, la RPL se determina implícitamente.

5. Un dispositivo informático que implementa un decodificador de video, en donde el dispositivo informático está adaptado para realizar un método según la reivindicación 4.

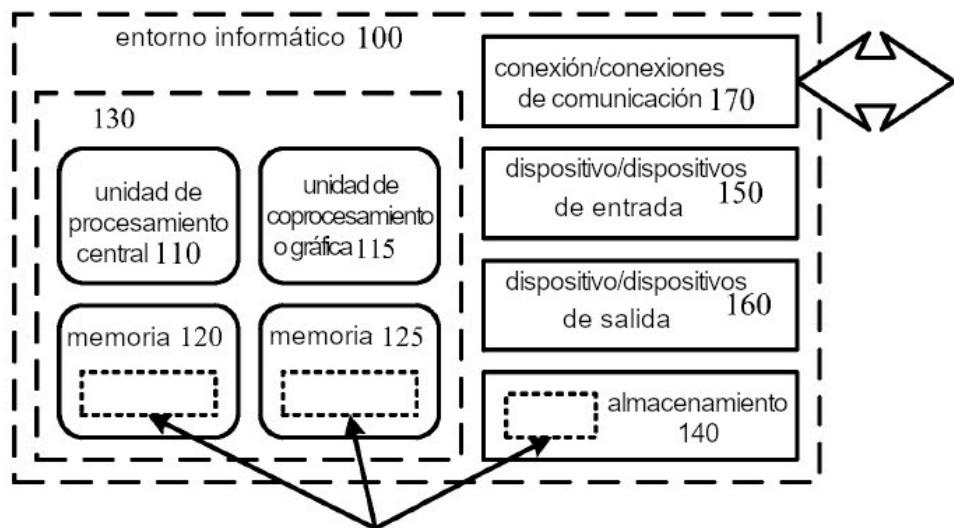
35 6. Uno o más medios legibles por ordenador que tienen almacenadas en el mismo instrucciones ejecutables por ordenador para hacer que una unidad de procesamiento programada de ese modo lleve a cabo un método que comprende:

40 evaluar (1010; 1110) si un valor de una variable que indica un número total de imágenes de referencia es mayor que 1; y

solo cuando el valor es mayor que 1, señalizar (1020) o analizar (1120) condicionalmente de un flujo de bits una estructura de modificación de una lista de imágenes de referencia, RPL, en donde la estructura de modificación de RPL incluye una bandera que indica si una RPL se modifica según elementos de sintaxis señalados explícitamente en el flujo de bits como una lista de entradas de lista,

45 en donde cada una de las entradas de la lista especifica un índice de una imagen de referencia en una versión temporal de la RPL que se colocará en una posición actual de la RPL,

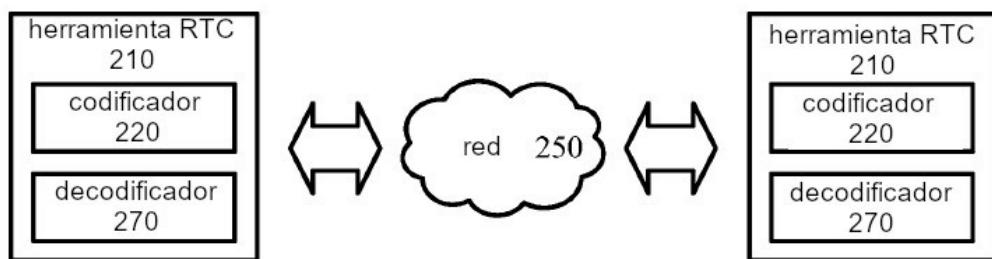
en donde si el valor de dicha bandera es igual a 1, la RPL se especifica explícitamente como la lista, y si el valor de dicha bandera es igual a 0, la RPL se determina implícitamente.



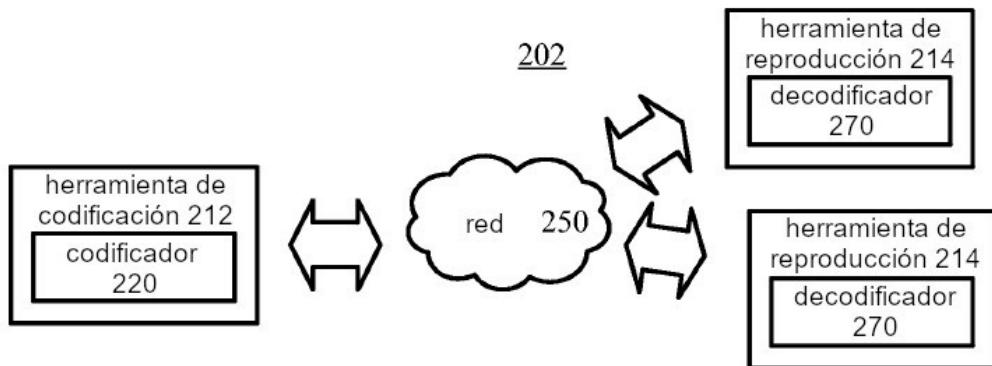
software 180 que implementa una o más innovaciones para señalización de información de modificación de lista de imágenes de referencia

## Figura 1

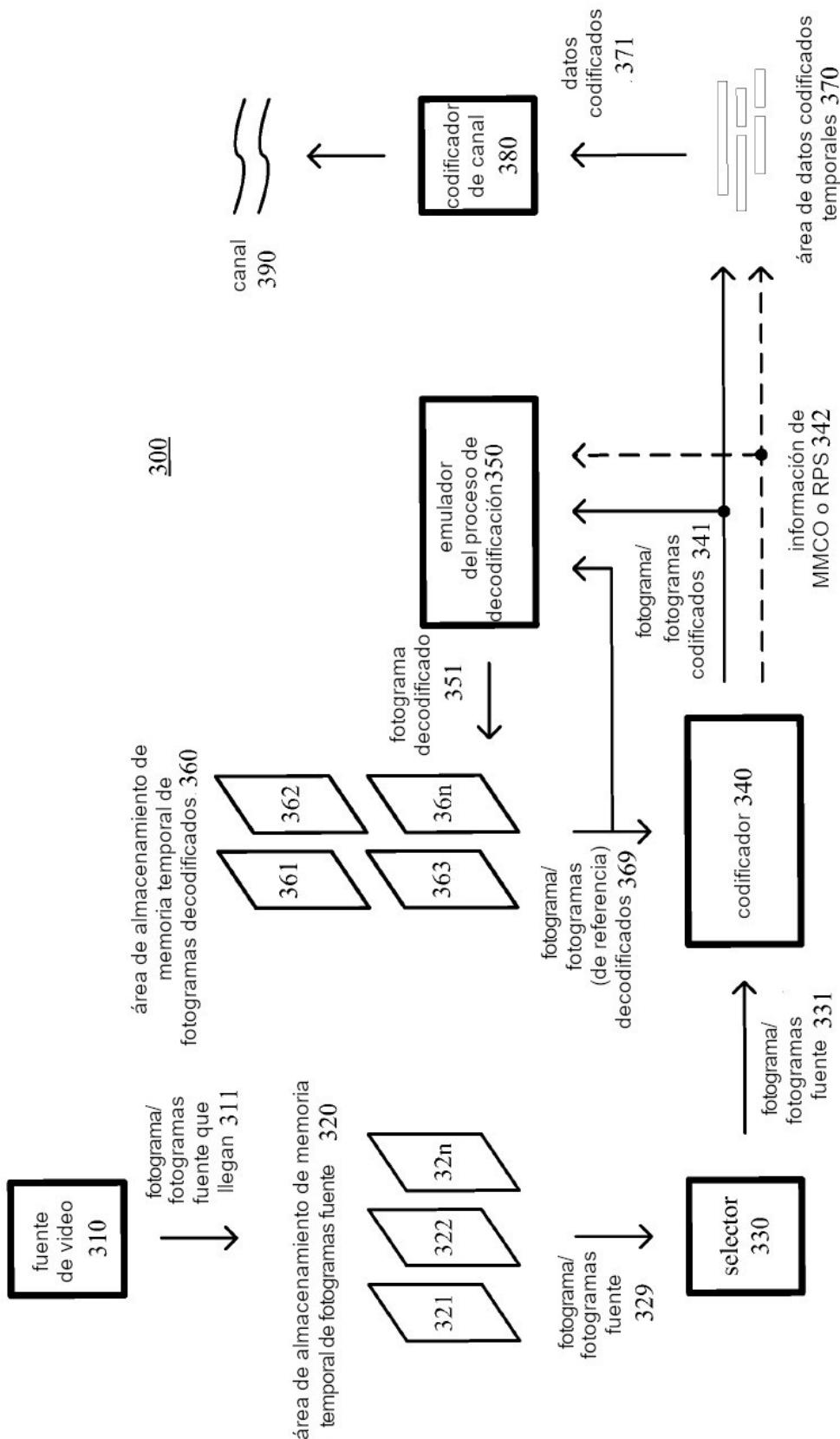
201



## Figura 2a



## Figura 2b

**Figura 3**

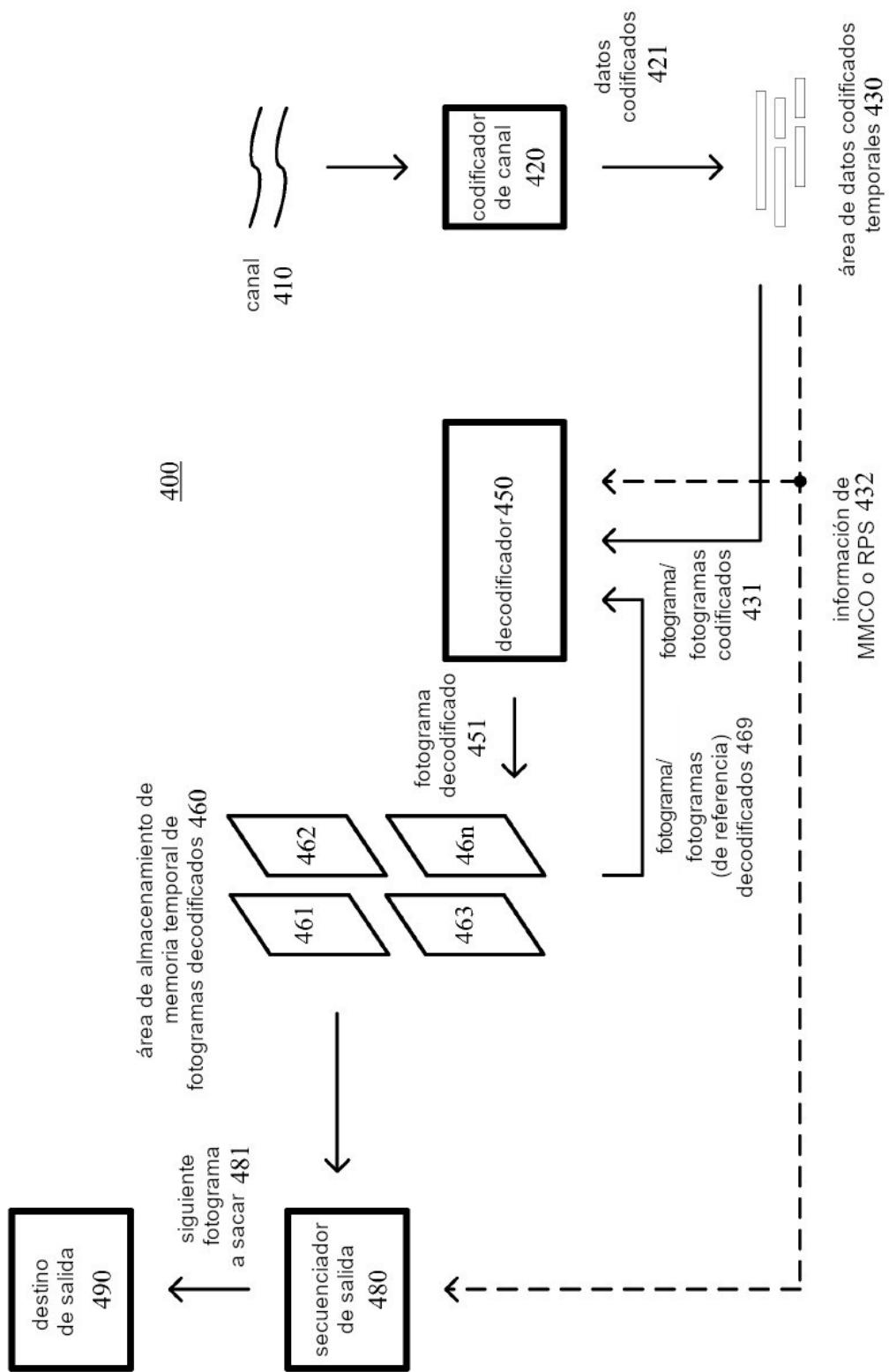


Figura 4

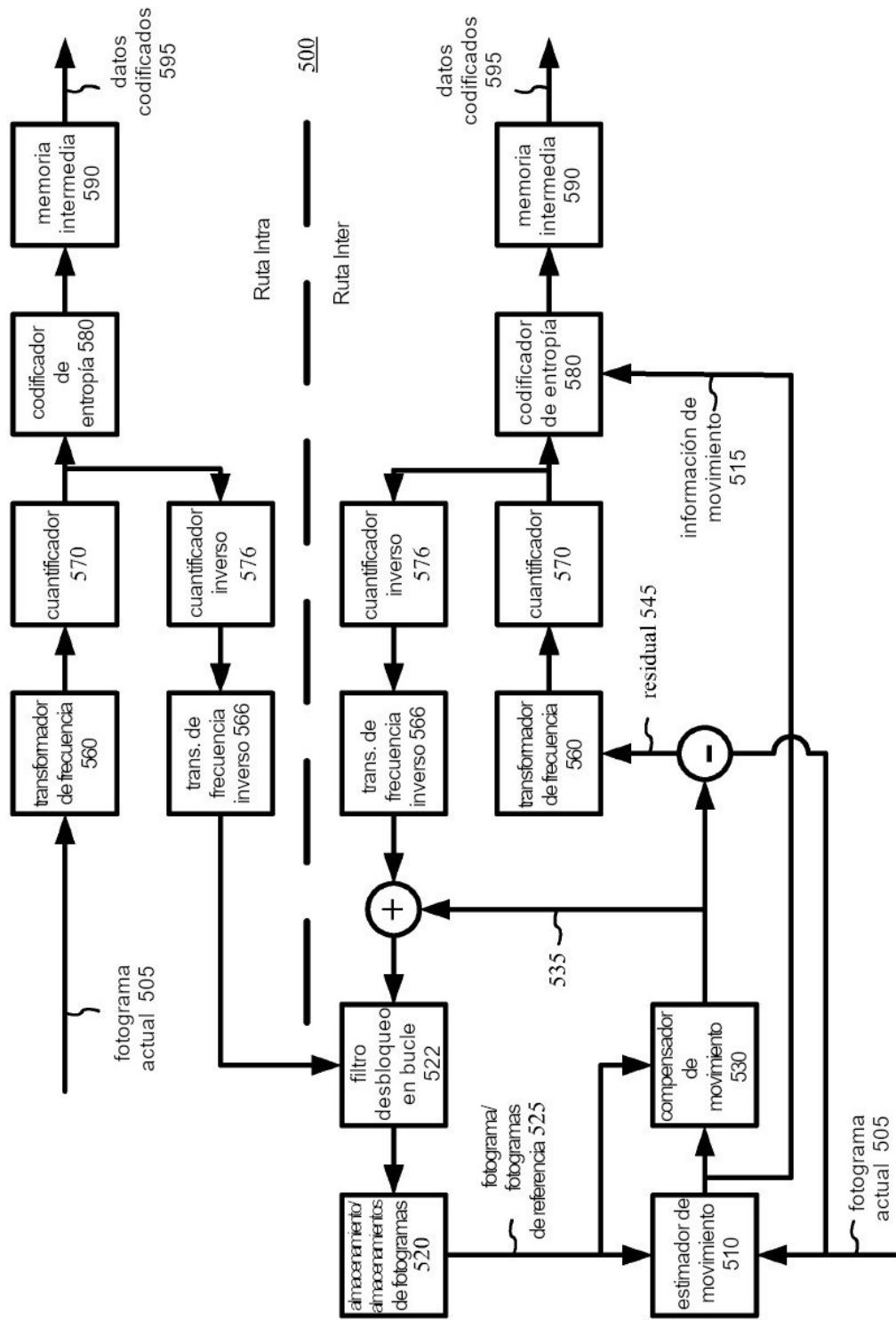
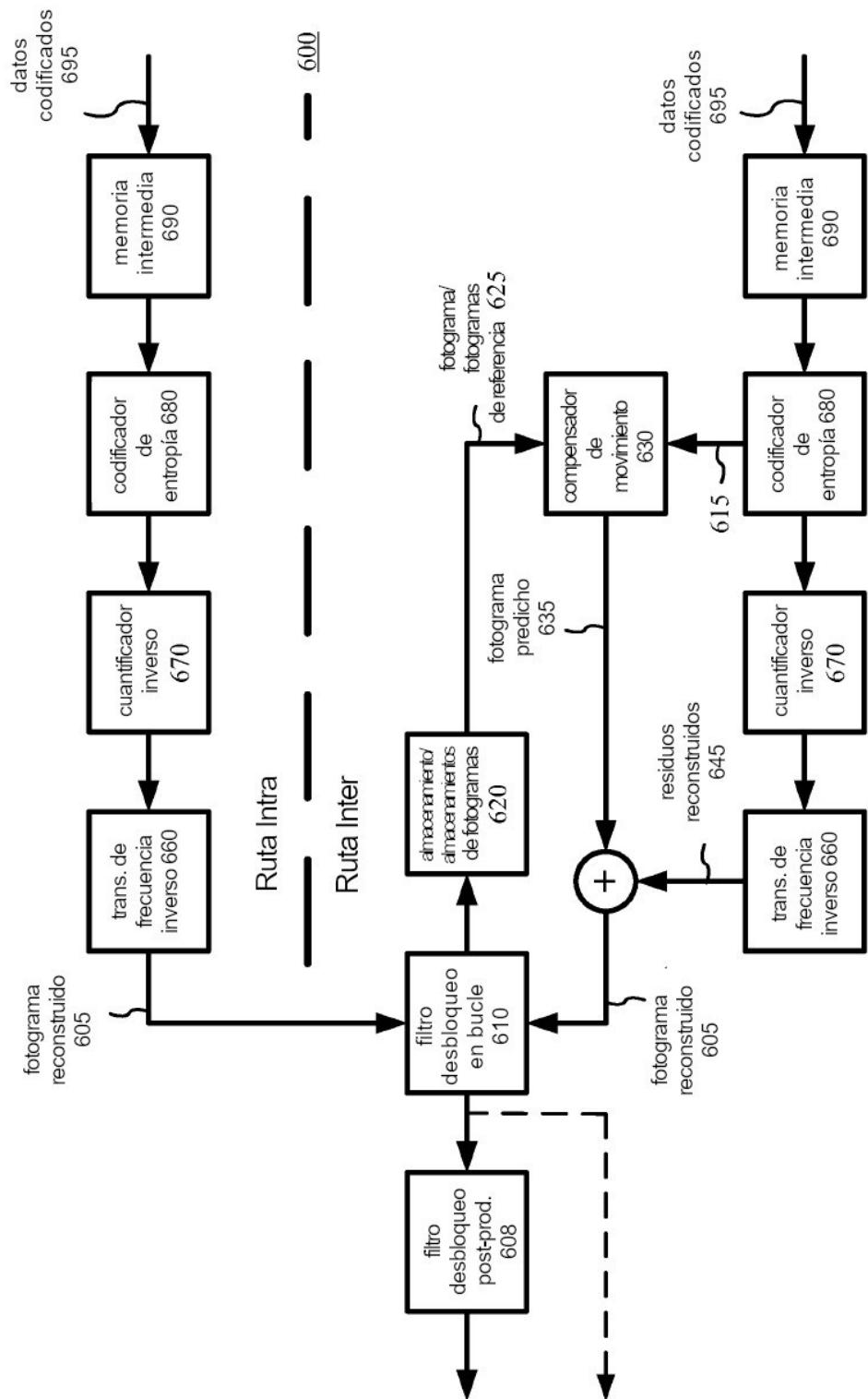


Figura 5

**Figura 6**

modificación_listas_imag_ref()	Descriptor
si(NumPocTotalAct > 1)	
bandera_modificación_lista_imag_ref_l0	u(1)
si (bandera_modificación_lista_imag_ref_l0)	
para (i=0; i<=num_ref_idx_l0_activo_menos1; i++)	
entrada_lista_l0 [i]	u(v)
si (tipo_segmento == B)	
si(NumPocTotalAct > 1)	
bandera_modificación_lista_imag_ref_l1	u(1)
si (bandera_modificación_lista_imag_ref_l1)	
para (i=0; i<=num_ref_idx_l1_activo_menos1; i++)	
entrada_lista_l1 [i]	u(v)
}	
}	

**Figura 7a**

750

	<b>Descriptor</b>
cabecera_segmento_segmento(){	
...	
si (tipo_segmento == P    tipo_segmento == B){	
bandera_anula_num_ref_idx_activo	u(1)
si (bandera_anula_num_ref_idx_activo){	
num_ref_idx_l0_activo_menos1	ue(v)
si (tipo_segmento == B)	
num_ref_idx_l1_activo_menos1	ue(v)
}	
si (bandera_presente_modificación_lista && NumPocTotalAct > 1)	
modificación_listas_imag_ref()	
...	
}	

**Figura 7b**760

	<b>Descriptor</b>
modificación_listas_imag_ref(){	
bandera_modificación_lista_imag_ref_l0	u(1)
si (bandera_modificación_lista_imag_ref_l0)	
para (i=0; i<=num_ref_idx_l0_activo_menos1; i++)	
entrada_lista_l0 []	u(v)
si (tipo_segmento == B){	
bandera_modificación_lista_imag_ref_l1	u(1)
si (bandera_modificación_lista_imag_ref_l1)	
para (i=0; i<=num_ref_idx_l1_activo_menos1; i++)	
entrada_lista_l1 []	u(v)
}	
}	

**Figura 7c**

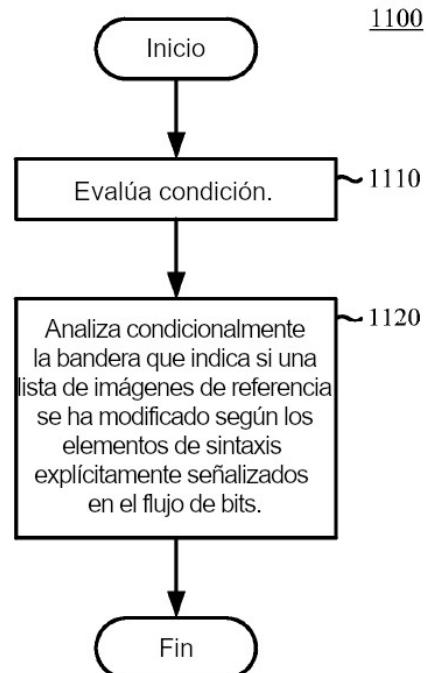
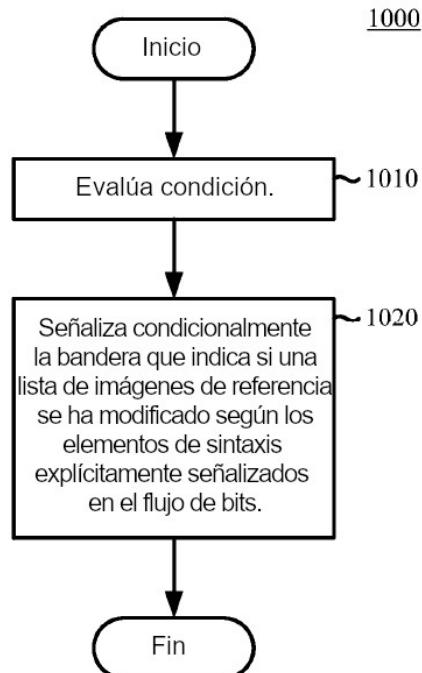
800

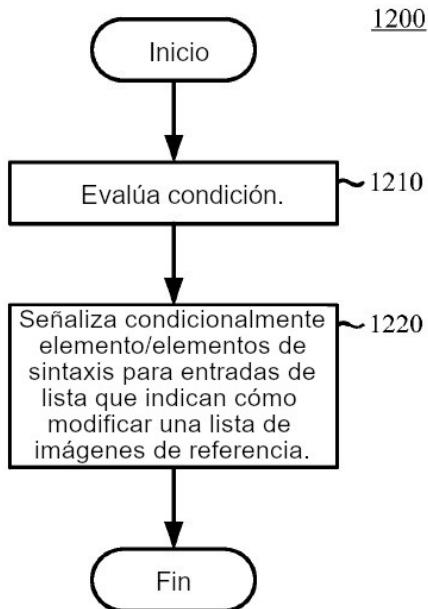
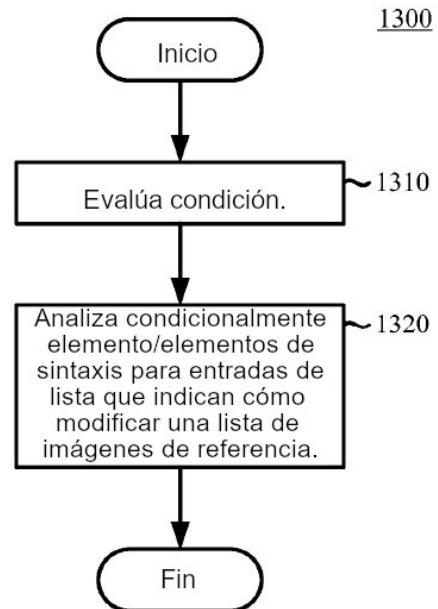
modificación_listas_imag_ref(){	Descriptor
<b>bandera_modificación_lista_imag_ref_l0</b>	u(1)
si (bandera_modificación_lista_imag_ref_l0 && !(NumPocTotalAct == 2 && num_ref_idx_l0_activo_menos1 == 0)) para (i=0; i<=num_ref_idx_l0_activo_menos1; i++)	
<b>entrada_lista_l0 [i]</b>	u(v)
si (tipo_segmento == B){	
<b>bandera_modificación_lista_imag_ref_l1</b>	u(1)
si (bandera_modificación_lista_imag_ref_l1 && !(NumPocTotalAct == 2 && num_ref_idx_l1_activo_menos1 == 0)) para (i=0; i<=num_ref_idx_l1_activo_menos1; i++)	
<b>entrada_lista_l1 [i]</b>	u(v)
}	
}	

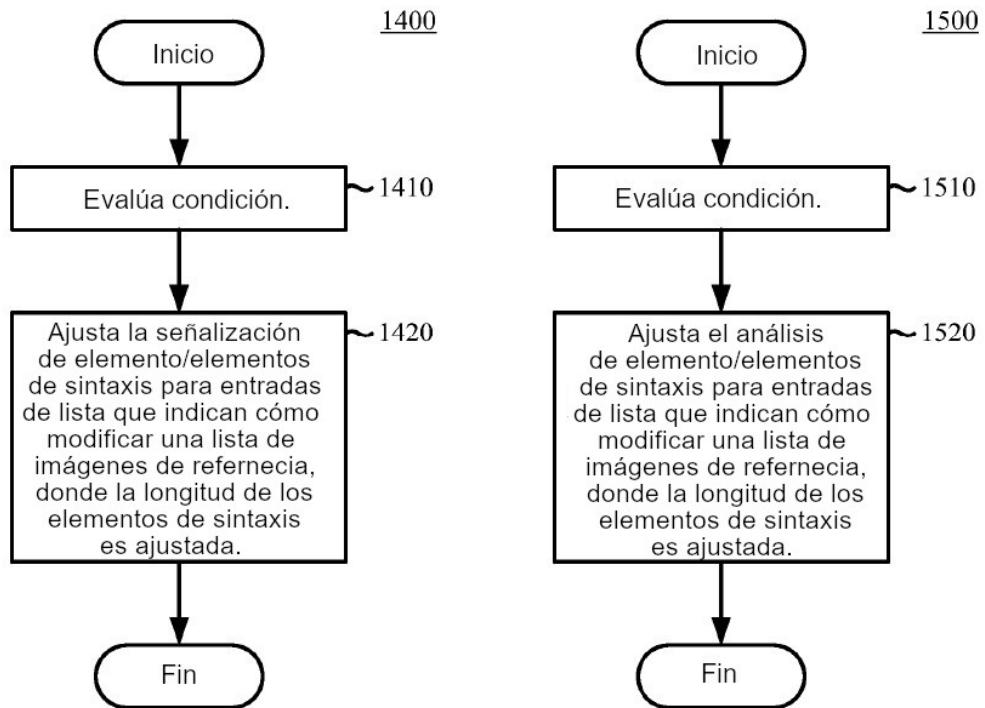
**Figura 8**900

modificación_listas_imag_ref(){	Descriptor
<b>bandera_modificación_lista_imag_ref_l0</b>	u(1)
si (bandera_modificación_lista_imag_ref_l0 && !(NumPocTotalAct == 2 && num_ref_idx_l0_activo_menos1 == 0) && !(NumPocTotalAct == 2 && num_ref_idx_l0_activo_menos1 == 1) && ((bandera_pred ponderada != 1 && tipo_segmento == P )    (bandera_bipred ponderada != 1 && tipo_segmento == B))) para (i=0; i<=num_ref_idx_l0_activo_menos1; i++)	
<b>entrada_lista_l0 [i]</b>	u(v)
si (tipo_segmento == B){	
<b>bandera_modificación_lista_imag_ref_l1</b>	u(1)
si (bandera_modificación_lista_imag_ref_l1 && !(NumPocTotalAct == 2 && num_ref_idx_l1_activo_menos1 == 0) && !(NumPocTotalAct == 2 && num_ref_idx_l1_activo_menos1 == 1) && (bandera_bipred ponderada != 1)) para (i=0; i<=num_ref_idx_l1_activo_menos1; i++)	
<b>entrada_lista_l1 [i]</b>	u(v)
}	
}	

**Figura 9**

**Figura 10****Figura 11**

**Figura 12****Figura 13**



**Figura 14**

**Figura 15**