

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **3 010 314**

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

96 Fecha de presentación y número de la solicitud europea: **23.09.2013** **E 21155916 (6)**

97 Fecha y número de publicación de la concesión europea: **06.11.2024** **EP 3840383**

54 Título: **Señalización condicional de información de modificación de la 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:  
**02.04.2025**

73 Titular/es:

**MICROSOFT TECHNOLOGY LICENSING, LLC**  
**(100.00%)**  
**One Microsoft Way**  
**Redmond, WA 98052-6399, US**

72 Inventor/es:

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

74 Agente/Representante:

**ISERN JARA, Jorge**

ES 3 010 314 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 la lista de imágenes de referencia

5 Antecedentes

Los ingenieros utilizan la compresión (también llamada codificación de fuente o codificación de fuente) para reducir la tasa de bits del vídeo digital. La compresión reduce el coste de almacenamiento y transmisión de información de vídeo al convertir la información a un formato 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/descodificador.

En las últimas dos décadas, se han adoptado diversos estándares de códecs de vídeo, 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, se está desarrollando el estándar HEVC. Un estándar de códec de vídeo normalmente define opciones para la sintaxis de un flujo de bits de vídeo codificado, detallando parámetros en el flujo de bits cuando se utilizan características particulares en la codificación y decodificación. En muchos casos, un estándar de códec de vídeo también proporciona detalles sobre las operaciones de decodificación que un decodificador debe realizar para lograr resultados correctos en la decodificación. Además de los estándares de códec, diversos formatos de códec propietarios definen otras opciones para la sintaxis de un flujo de bits de vídeo codificado y las operaciones de decodificación correspondientes.

Algunos tipos de parámetros en un flujo de bits indican información sobre las 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 utilizarse para la predicción interimagen en el proceso de decodificación de otras imágenes. Normalmente, las demás imágenes siguen a 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 códecs de vídeo, hay múltiples imágenes de referencia disponibles en un momento determinado para su uso en la predicción con compensación de movimiento. Estos estándares/formatos 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 utilizada para la predicción con compensación de movimiento. En algunos formatos y estándares de códecs de vídeo, un conjunto de imágenes de referencia ("RPS") es un conjunto de imágenes de referencia disponible para su uso en la predicción con compensación de movimiento en un momento determinado, y una RPL son algunas de las imágenes de referencia en el RPS. Las imágenes de referencia en una RPL se direccionan con índices de referencia. Un índice de referencia identifica una imagen de referencia en la RPL. Durante la codificación y decodificación, se puede actualizar un RPS para tener en cuenta las imágenes recientemente 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 tal manera que las imágenes de referencia más comúnmente utilizadas se identifiquen con índices de referencia que sean más eficientes para señalar. En algunos estándares de códecs recientes, se construye una RPL durante la codificación y la decodificación basándose en la información disponible sobre el RPS, modificaciones de acuerdo con las reglas y/o modificaciones señalizadas en el flujo de bits. La señalización de modificaciones para una RPL puede consumir una cantidad significativa de bits.

BROSS B ET AL: "High Efficiency Video Coding (HEVC) text specification draft 8", 10. JCT\_VC MEETING; 101. MPEG MEETING especifica la codificación de vídeo de alta eficiencia. En particular, se divulga la siguiente sintaxis: `if(lists_modification_present_flag)`

`ref_pic_list_modification()`. La sección 7.3.3.3 de dicho documento divulga una sintaxis de modificación de la lista de imágenes de referencia.

Por lo tanto, el objeto de la presente invención es proporcionar un método mejorado en un dispositivo informático que implementa un decodificador de vídeo, un sistema informático correspondiente que implementa un sistema codificador de vídeo y un medio legible por ordenador correspondiente.

Este objeto se resuelve mediante la materia objeto de las reivindicaciones independientes.

Las realizaciones preferidas están definidas por las reivindicaciones dependientes.

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 información de modificación de la RPL cuando no se utilizaría o cuando se pueden inferir valores de dicha información.

De acuerdo con un aspecto de las innovaciones descritas en este documento, un codificador de vídeo evalúa una condición. Dependiendo de los resultados de la evaluación, el codificador señala condicionalmente en un flujo de bits un indicador que indica si una RPL se modifica de acuerdo con elementos de sintaxis señalados explícitamente en el flujo de bits. Un decodificador de vídeo correspondiente evalúa una condición. Dependiendo de los resultados de la evaluación, el decodificador analiza condicionalmente a partir de un flujo de bits un indicador que indica si una RPL se modifica de acuerdo con elementos de sintaxis señalados explícitamente en el flujo de bits. En algunas implementaciones de ejemplo, la RPL puede ser para un segmento predictivo ("P") o un segmento bipredictivo ("B"). Alternativamente, una estructura de sintaxis de nivel superior se señala/analiza condicionalmente basado en la evaluación de la condición.

En algunas implementaciones de ejemplo, si no se modifica la RPL, se construye una RPL predeterminada basado en las reglas sobre la construcción de la 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.

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 de ejemplo, la condición es si el valor de la variable es mayor que 1.

La condición se puede evaluar como parte del procesamiento de una estructura de modificación de la RPL que incluye el indicador. O, la condición se puede evaluar como parte del procesamiento de un encabezado de segmento, en cuyo caso la estructura de modificación de la RPL (incluido el indicador) se señala o analiza condicionalmente dependiendo de los resultados de la evaluación.

De acuerdo con otro aspecto de las innovaciones descritas en este documento, un codificador de vídeo evalúa otra condición. Dependiendo de los resultados de la evaluación, el codificador señala condicionalmente en un flujo de bits uno o más elementos de sintaxis para entradas de la lista que indican cómo modificar una RPL (por ejemplo, reemplace la RPL, ajuste la RPL). Un decodificador de vídeo correspondiente evalúa la condición. Dependiendo de los resultados de la evaluación, el decodificador analiza condicionalmente a partir de un flujo de bits uno o más elementos de sintaxis en busca de entradas de la lista que indican cómo modificar una RPL. (por ejemplo, reemplazar la RPL, ajustar la RPL). En algunas implementaciones de ejemplo, 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 una 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 deshabilitada. Se puede utilizar 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 está señalando/analizando. En algunas implementaciones de ejemplo, 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 el 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 de ejemplo, 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.

De acuerdo con otro aspecto de las innovaciones descritas en el presente documento, un codificador de vídeo 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, reemplazar la RPL, ajustar 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 vídeo 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 se ajusta la longitud (en bits) de al menos uno de los uno o más elementos de sintaxis). Por ejemplo, la condición depende al menos en parte de si la predicción ponderada está deshabilitada. Se puede utilizar 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 está señalando/analizando. En algunas implementaciones de 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. Por ejemplo, en algunas implementaciones de ejemplo, si la predicción

ponderada está deshabilitada, la longitud de un elemento de sintaxis determinado para la entrada de la lista[i] es  $\text{Ceil}(\text{Log2}(\text{NumPocTotalCurr}-i))$  bits. Por otro lado, si se habilita la predicción ponderada, la longitud del elemento de sintaxis determinado para la entrada de la lista[i] es  $\text{Ceil}(\text{Log2}(\text{NumPocTotalCurr}))$  bits.

5 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.

10 Los anteriores y otros objetos, características y ventajas de la invención se harán más evidentes a partir de la siguiente descripción detallada, que se hace con referencia a las figuras adjuntas.

Breve descripción de los dibujos

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

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

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

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

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

30 La Figura 6 es un diagrama que ilustra un decodificador de vídeo de ejemplo junto con el cual se pueden implementar algunas realizaciones descritas.

La Figura 7a es una tabla que ilustra la señalización condicional de un indicador que indica si se modifica una RPL, de acuerdo con algunas implementaciones de ejemplo.

35 Las Figuras 7b y 7c son tablas que ilustran la señalización condicional de uno o más indicadores que indican si se modifica una RPL, de acuerdo con otras implementaciones de ejemplo.

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

40 Las Figuras 10 y 11 son diagramas de flujo que ilustran técnicas generalizadas para la señalización y el análisis, condicional respectivamente, de un indicador que indica si se modifica una RPL.

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

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

Descripción detallada

55 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 la RPL cuando no se utilizaría o cuando se pueden inferir valores de dicha información.

60 En algunos estándares de códecs 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 bipredictivo ("B"), hay dos RPL, que se denominan RPL 0 y RPL 1. Al comienzo del proceso de decodificación de 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 de acuerdo con reglas y/o modificaciones señalizadas 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 (tal como el conjunto de imágenes de referencia disponibles en el decodificador para la decodificación de la imagen actual), modificaciones de acuerdo con reglas y/o modificaciones señalizadas en el flujo de bits. De manera más general, una RPL se construye durante la codificación y la decodificación basándose en la información disponible sobre la RPL, las modificaciones de acuerdo con reglas y/o las modificaciones señalizadas en el flujo de bits. La señalización de modificaciones para una RPL puede consumir una cantidad significativa de bits. En algunos estándares de códecs recientes, existen ineficiencias en la forma en que se señala la información de modificación de la RPL.

La descripción detallada presenta diversas innovaciones en el área de señalización de información de modificación de la RPL. En algunas situaciones, estas innovaciones dan como resultado una señalización más eficiente de los elementos de sintaxis para la información de modificación de la RPL. Por ejemplo, la descripción detallada describe la señalización condicional de elementos de sintaxis para entradas de la 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 un indicador que indica si se modifica una RPL.

En algunas implementaciones de ejemplo, si no se modifica la RPL, se construye una RPL predeterminada de acuerdo con un enfoque "implícito" utilizando reglas sobre la construcción de la RPL a partir de un RPS. Si se modifica la RPL, se construye una RPL de reemplazo de acuerdo con un enfoque de señalización "explícito" utilizando información de modificación de la 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 relativos al RPL predeterminado.

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, 9th meeting of the Joint Collaborative Team on Video Coding ("JCTVC"), Geneva, April 2012. Véase también la versión preliminar titulada "High Efficiency Video Coding (HEVC) text specification draft 9," JCTVC-K1003\_d11, 11th meeting of the JCT-VC, Shanghai, October 2012. Las innovaciones descritas en este documento también pueden implementarse para otros estándares o formatos.

De manera más general, son posibles diversas alternativas a los ejemplos descritos en este documento. Por ejemplo, algunos de los métodos descritos en este documento se pueden alterar cambiando el orden de los acciones del método descrito, dividiendo, repitiendo u omitiendo ciertos acciones del método, etc. Los diversos aspectos de la tecnología divulgada se pueden utilizar 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 mencionados en los antecedentes. Normalmente, una técnica o herramienta determinada no resuelve todos estos problemas.

#### I. Sistemas informáticos de ejemplo.

La Figura 1 ilustra un ejemplo generalizado de un sistema informático (100) adecuado en el que se pueden implementar varias de las innovaciones descritas. El sistema informático (100) 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 de propósito especial.

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

Un sistema informático puede tener características adicionales. Por ejemplo, el sistema informático (100) incluye almacenamiento (140), uno o más dispositivos de entrada (150), uno o más dispositivos de salida (160)

y una o más conexiones de comunicación (170). Un mecanismo de interconexión (no mostrado), como un bus, un controlador o una red, interconecta los componentes del sistema informático (100). Normalmente, el software del sistema operativo (no mostrado) proporciona un entorno operativo para otro software que se ejecuta en el sistema informático (100) y coordina las actividades de los componentes del sistema informático (100).

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

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

La(s) conexión(es) de comunicación (170) permite(n) la comunicación a través de un medio de comunicación con otra entidad informática. El medio de comunicación transmite información como instrucciones ejecutables por ordenador, entrada o salida de audio o vídeo 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 modificadas de tal manera que se codifica la información en la señal. A modo de ejemplo, y sin limitación, los medios de comunicación pueden utilizar un portador eléctrico, óptico, de RF u otro.

Las innovaciones pueden describirse en el contexto general de los medios legibles por ordenador. Los medios legibles por ordenador son cualquier medio tangible disponible al que se pueda acceder dentro de un entorno informático. A modo de ejemplo, y sin limitación, con el sistema informático (100), los medios legibles por ordenador incluyen memoria (120, 125), almacenamiento (140) y combinaciones de cualquiera de los anteriores.

Las innovaciones pueden describirse en el contexto general de instrucciones ejecutables por ordenador, como las incluidas en módulos de programa, que se ejecutan en un sistema informático en un procesador real o virtual de destino. 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 del programa se puede combinar o dividir entre los módulos del programa según se desee en diversas 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 utilizan indistintamente en este documento. A menos que el contexto indique claramente lo contrario, ninguno de los términos implica limitación alguna respecto de un tipo de sistema informático o dispositivo informático. En general, un sistema informático o un 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 descrita en este documento.

Los métodos divulgados también pueden implementarse utilizando hardware informático especializado configurado para realizar cualquiera de los métodos divulgados. Por ejemplo, los métodos divulgados pueden implementarse mediante un circuito integrado (por ejemplo, un circuito integrado de aplicación específica ("ASIC") (tal como una unidad de procesamiento de señal digital ("DSP") ASIC, una unidad de procesamiento de gráficos ("GPU") o un dispositivo lógico programable ("PLD"), tal como una matriz de puertas programables en campo ("FPGA")) especialmente diseñado o configurado para implementar cualquiera de los métodos divulgados.

Para facilitar la presentación, la descripción detallada utiliza términos como "determinar" y "utilizar" para describir las operaciones informáticas 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 dependiendo de la implementación.

II. Entornos de red de ejemplo.

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

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

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

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

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

### III. Sistemas codificadores de ejemplo.

La Figura 3 es un diagrama de bloques de un sistema codificador (300) de ejemplo junto con el cual se pueden implementar algunas realizaciones descritas. El sistema codificador (300) puede ser una herramienta de codificación de propósito general capaz de operar en cualquiera de múltiples modos de codificación, tales como un modo de codificación de baja latencia para comunicación en tiempo real, un modo de transcodificación y un 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 uno de dichos modos de codificación. El sistema codificador (300) se puede implementar como un módulo de sistema operativo, como parte de una biblioteca de aplicaciones o como una aplicación independiente. En general, el sistema codificador (300) recibe una secuencia de fotogramas de vídeo de origen (311) desde una fuente de vídeo (310) 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 la RPL.

La fuente de vídeo (310) puede ser una cámara, una tarjeta sintonizadora, un medio de almacenamiento u otra fuente de vídeo digital. La fuente de vídeo (310) produce una secuencia de fotogramas de vídeo a una tasa de fotogramas de, por ejemplo, 30 fotogramas por segundo. Tal como se utiliza en este documento, el término "fotograma" generalmente se refiere a datos de una imagen origen, codificados o reconstruidos. Para el vídeo progresivo, un fotograma es un fotograma de vídeo progresivo. Para el vídeo entrelazado, en realizaciones de ejemplo, un fotograma de vídeo entrelazado se desentrelaza antes de la codificación. Alternativamente, dos campos de vídeo entrelazados complementarios se codifican como un fotograma de vídeo entrelazado o

campos separados. Además de indicar un fotograma de vídeo progresivo, el término "fotograma" o "imagen" puede indicar un único campo de vídeo no emparejado, un par complementario de campos de vídeo, un plano de objeto de vídeo que representa un objeto de vídeo en un momento determinado o una región de interés en una imagen más grande. El plano o región del objeto de vídeo puede ser parte de una imagen más grande que incluye múltiples objetos o regiones de una escena.

Un fotograma de origen (311) entrante se almacena en un área de almacenamiento de memoria temporal de fotogramas de origen (320) que incluye múltiples áreas de almacenamiento de búfer de fotograma (321, 322, ..., 32n). Un búfer de fotograma (321, 322, etc.) contiene un fotograma de origen en el área de almacenamiento de fotogramas de origen (320). Después de que uno o más de los fotogramas de origen (311) se hayan almacenado en búferes de fotogramas (321, 322, etc.), un selector de fotogramas (330) selecciona periódicamente un fotograma de origen individual del área de almacenamiento de fotogramas de origen (320). El orden en el que el selector de fotogramas (330) selecciona los fotogramas para su entrada al codificador (340) puede diferir del orden en el que la fuente de vídeo (310) produce los fotogramas, por ejemplo, un fotograma puede estar adelante en el orden, para facilitar la predicción temporalmente hacia atrás. Antes del codificador (340), el sistema codificador (300) 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 seleccionado (331) para producir un fotograma codificado (341) y también produce señales de operación de control de gestión de memoria ("MMCO") (342) 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 utilizar uno o más fotogramas previamente codificados/decodificados (369) que han sido almacenados en un área de almacenamiento de memoria temporal de fotograma decodificado (360). Dichos fotogramas decodificados (369) almacenados se utilizan como imágenes de referencia para la predicción interfotograma del contenido del fotograma de origen (331) 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, cuantización 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 Windows Media Vídeo, formato VC-1, formato MPEG-x (por ejemplo, MPEG-1, MPEG-2 o MPEG-4), formato H.26x (por ejemplo, H.261, H.262, H.263, H.264), formato HEVC u otro formato.

Por ejemplo, dentro del codificador (340), un fotograma predicho e 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 de origen (341) con respecto a uno o más fotogramas de referencia (369). 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 uno o más RPL, con índices de referencia que abordan las imágenes de referencia en las RPL. Las RPL se construyen durante la codificación para que se agreguen nuevas imágenes de referencia cuando sea apropiado, se eliminen imágenes de referencia más antiguas que ya no se usan para la compensación de movimiento cuando sea apropiado y se reordenen las imágenes de referencia cuando sea apropiado. En algunas implementaciones, por ejemplo, al codificar una imagen actual, el codificador (340) determina un RPS que incluye imágenes de referencia en el área de almacenamiento de fotograma decodificado (360), luego crea una o más RPL para la codificación de un segmento determinado de la imagen actual. Se puede crear una RPL aplicando reglas sobre la selección de imágenes de referencia disponibles en el RPS (enfoque implícito), en cuyo caso la información de modificación de la RPL no se señala explícitamente en el flujo de bits. O bien, la RPL se puede crear seleccionando imágenes de referencia específicas disponibles en el RPS, donde las imágenes de referencia que se seleccionan se indicarán en la información de modificación de la 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. Como alternativa, la información de modificación de la 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 imágenes de referencia en la RPL construido mediante reglas del enfoque implícito.

Al codificar un fotograma intercodificado, el codificador (340) puede evaluar los resultados de la compensación de movimiento para los cuales no se modifica una RPL de acuerdo con elementos de sintaxis explícitamente señalados en el flujo de bits, y también evaluar los resultados de la compensación de movimiento para los cuales se modifica la RPL de acuerdo con elementos de sintaxis explícitamente señalados 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 la RPL señalizada en el flujo de bits) o una RPL modificada (con información de modificación de la 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 uno o más de (a) reordenar imágenes de referencia para un direccionamiento más eficiente con índices de referencia, (b) eliminar imágenes de referencia basado al menos en parte en la frecuencia de uso durante la codificación, y (c) agregar imágenes de referencia basado al menos en parte en la frecuencia de uso durante la codificación. Por ejemplo, el codificador (340) puede decidir eliminar una imagen de referencia determinada 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 de acuerdo con otros criterios. Como otro ejemplo, el codificador (340) puede decidir agregar una imagen de referencia determinada a la RPL si la utilización de la imagen de referencia para la compensación de movimiento es superior a una cantidad umbral y/o de acuerdo con otros criterios. Como otro ejemplo, el codificador (340) puede decidir cómo reordenar las imágenes de referencia en la RPL basado en la frecuencia de utilización de las respectivas imágenes de referencia y/o de acuerdo con otros criterios.

El estimador de movimiento genera información de movimiento, como información del 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 con compensación de 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 además mediante una transformación de frecuencia, cuantización y codificación de entropía. De manera similar, para la intrapredicción, el codificador (340) puede determinar valores de intrapredicción para un bloque, determinar valores residuales de predicción y codificar los valores residuales de predicción (con una transformación de frecuencia, cuantización y codificación de entropía). En particular, el codificador de entropía del codificador (340) comprime los valores de los coeficientes de transformación cuantizados, así como cierta información lateral (por ejemplo, información del vector de movimiento, valores QP, decisiones de modo, opciones de parámetros, índices de referencia, información de modificación de la 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"), codificación de longitud variable a longitud fija ("V2F"), codificación LZ, codificación de diccionario, codificación de entropía de partición de intervalo de probabilidad ("PIPE") y combinaciones de las anteriores. El codificador de entropía puede utilizar 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 codificados (341) y la información MMCO/RPS (342) son procesados por un emulador de proceso de decodificación (350). El emulador del proceso de decodificación (350) implementa algunas de las funcionalidades de un decodificador, por ejemplo, tareas de decodificación para reconstruir imágenes de referencia que son utilizadas por el codificador (340) en la compensación de movimiento. El emulador del proceso de decodificación (350) utiliza la información MMCO/RPS (342) para determinar si un fotograma codificado (341) determinado necesita ser reconstruido y almacenado para su uso como una imagen de referencia en la predicción interfotograma de los fotogramas subsiguientes que se codificarán. Si la información MMCO/RPS (342) indica que es necesario almacenar un fotograma codificado (341), el emulador del proceso de decodificación (350) modela el proceso de decodificación que llevaría a cabo un decodificador que recibe el fotograma codificado (341) y produce un fotograma decodificado (351) correspondiente. Al hacerlo, cuando el codificador (340) ha utilizado el(los) fotograma(s) decodificado(s) (369) que se han almacenado en el área de almacenamiento de fotograma decodificado (360), el emulador del proceso de decodificación (350) también utiliza el(los) fotograma(s) decodificado(s) (369) del área de almacenamiento (360) como parte del proceso de decodificación.

El área de almacenamiento de memoria temporal del fotograma decodificado (360) incluye múltiples áreas de almacenamiento de búfer de fotograma (361, 362, ..., 36n). El emulador de proceso de decodificación (350) utiliza la información MMCO/RPS (342) para gestionar los contenidos del área de almacenamiento (360) con el fin de identificar cualquier búfer de fotogramas (361, 362, etc.) con fotogramas que ya no son necesarios para que el codificador (340) los utilice como imágenes de referencia. Después de modelar el proceso de decodificación, el emulador de proceso de decodificación (350) almacena un fotograma decodificado (351) recientemente en un búfer de fotogramas (361, 362, etc.) que ha sido identificado de esta manera.

Los fotogramas codificados (341) y la información MMCO/RPS (342) también se almacenan en búfer en un área de datos codificados (370) temporal. Los datos codificados que se agregan en el área de datos codificados (370) 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 la RPL. Los datos codificados que se agregan en el área de datos codificados (370) también pueden incluir metadatos multimedia relacionados con los datos de vídeo codificados. (por ejemplo, como uno o más parámetros en uno o más mensajes de información mejorados complementarios ("SEI") o mensajes de información de usabilidad de vídeo ("VUI")).

Los datos agregados (371) del área de datos codificados (370) temporales son procesados por un codificador de canal (380). El codificador de canal (380) puede empaquetar los datos agregados para su transmisión como

un flujo de medios (por ejemplo, de acuerdo con un formato de contenedor de medios como ISO/IEC 14496-12), en cuyo caso el codificador de canal (380) puede agregar elementos de sintaxis como parte de la sintaxis del flujo de transmisión de medios. O, el codificador de canal (380) puede organizar los datos agregados para almacenarlos como un archivo. (por ejemplo, de acuerdo con un formato de contenedor de medios como ISO/IEC 14496-12), en cuyo caso el codificador de canal (380) puede agregar elementos de sintaxis como parte de la sintaxis del archivo de almacenamiento de medios. O, más generalmente, el codificador de canal (380) puede implementar uno o más protocolos de multiplexación de sistemas de medios o protocolos de transporte, en cuyo caso el codificador de canal (380) puede agregar elementos de sintaxis como parte de la sintaxis del(de los) protocolo(s). El codificador de canal (380) proporciona salida a un canal (390), que representa almacenamiento, una conexión de comunicaciones u otro canal para la salida.

#### IV. Sistemas decodificadores de ejemplo.

La Figura 4 es un diagrama de bloques de un sistema decodificador (400) de ejemplo junto con el cual se pueden implementar algunas realizaciones descritas. El sistema decodificador (400) puede ser una herramienta de decodificación de propósito general capaz de operar en cualquiera de múltiples modos de decodificación, tales 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 uno de dichos modos de decodificación. El sistema decodificador (400) se puede implementar como un módulo de sistema operativo, como parte de una biblioteca de aplicaciones o como una aplicación independiente. En general, el sistema decodificador (400) recibe datos codificados de un canal (410) y produce fotogramas reconstruidos como salida para un destino de salida (490). Los datos codificados pueden incluir elementos de sintaxis que indican información de modificación de la RPL.

El sistema decodificador (400) 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 de canal (420) puede procesar los datos codificados. Por ejemplo, el decodificador de canal (420) desempaqueta los datos que se han agregado para su transmisión como un flujo de medios (por ejemplo, de acuerdo con un formato de contenedor de medios como ISO/IEC 14496-12), en cuyo caso el decodificador de canal (420) puede analizar elementos de sintaxis agregados como parte de la sintaxis del flujo de transmisión de medios. O, el decodificador de canal (420) separa los datos de vídeo codificados que se han agregado para su almacenamiento como un archivo. (por ejemplo, de acuerdo con un formato de contenedor de medios como ISO/IEC 14496-12), en cuyo caso el decodificador de canal (420) puede analizar elementos de sintaxis agregados como parte de la sintaxis del archivo de almacenamiento de medios. O, más generalmente, el decodificador de canal (420) puede implementar uno o más protocolos de demultiplexación de sistemas de medios o protocolos de transporte, en cuyo caso el decodificador de canal (420) puede analizar elementos de sintaxis agregados como parte de la sintaxis de los protocolos.

Los datos codificados (421) que salen del decodificador de canal (420) se almacenan en un área de datos codificados (430) temporal hasta que se haya recibido una cantidad suficiente de dichos datos. Los datos codificados (421) incluyen fotogramas codificados (431) e información MMCO/RPS (432). Los datos codificados (421) en el área de datos codificados (430) 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 la RPL. Los datos codificados (421) en el área de datos codificados (430) también pueden incluir metadatos multimedia relacionados con los datos de vídeo codificados. (por ejemplo, como uno o más parámetros en uno o más mensajes SEI o mensajes VUI). En general, el área de datos codificados (430) almacena temporalmente datos codificados (421) hasta que dichos datos codificados (421) son utilizados por el decodificador (450). En ese punto, los datos codificados para un fotograma codificado (431) y la información MMCO/RPS (432) se transfieren desde el área de datos codificados (430) al decodificador (450). A medida que continúa la decodificación, se agregan nuevos datos codificados al área de datos codificados (430) y los datos codificados más antiguos que permanecen en el área de datos codificados (430) se transfieren al decodificador (450).

El decodificador (450) decodifica periódicamente un fotograma codificado (431) para producir un fotograma decodificado (451) correspondiente. Según corresponda, al realizar su proceso de decodificación, el decodificador (450) puede utilizar uno o más fotogramas decodificados (469) previamente como fotogramas de referencia (imágenes de referencia) para la predicción interfotograma. El decodificador (450) lee dichos fotogramas decodificados (469) previamente desde un área de almacenamiento de memoria temporal de fotograma decodificado (460). Generalmente, el decodificador (450) incluye múltiples módulos de decodificación que realizan tareas de decodificación tales como decodificación de entropía, cuantización inversa, transformadas de frecuencia inversa y compensación de movimiento (que pueden crear RPL utilizando información de modificación de la RPL). Las operaciones exactas realizadas por el decodificador (450) pueden variar dependiendo del formato de compresión.

Por ejemplo, el decodificador (450) recibe datos codificados para un fotograma comprimido o una secuencia de fotogramas y produce una salida que incluye el fotograma decodificado (451). En el decodificador (450), un

búfer 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 datos cuantizados codificados por entropía, así como información lateral codificada por entropía (incluidos índices de referencia, información de modificación de la RPL, etc.), aplicando normalmente la inversa de la codificación de entropía realizada en el codificador. El decodificador construye una o más RPL para imágenes de referencia, con índices de referencia que abordan las imágenes de referencia en las RPL. Las RPL se construyen de modo que se agreguen nuevas imágenes de referencia cuando sea apropiado, se eliminen las imágenes de referencia más antiguas que ya no se utilizan para la compensación de movimiento cuando sea apropiado y se reordenen las imágenes de referencia cuando sea apropiado. En algunas implementaciones, por ejemplo, al decodificar una imagen actual, el decodificador (450) determina un RPS que incluye imágenes de referencia en el área de almacenamiento de fotograma decodificado (460), luego crea una o más RPL para la decodificación de un segmento determinado de la imagen actual. Se puede crear una RPL aplicando 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 desde el flujo de bits. O, la RPL se puede crear seleccionando imágenes de referencia específicas disponibles en el RPS, donde las imágenes de referencia que se seleccionan se indican en la información de modificación de la RPL que se analiza desde 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 la 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 imágenes de referencia en la RPL construida mediante 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 subbloques y/o bloques (generalmente, bloques) del fotograma que se está reconstruyendo. Un módulo de intrapredicción puede predecir espacialmente valores de muestra de un bloque actual a partir de valores de muestra vecinos reconstruidos previamente. El decodificador (450) también reconstruye los residuos de predicción. Un cuantizador inverso cuantiza inversamente los datos decodificados por entropía. Un transformador de frecuencia inversa convierte los datos del dominio de frecuencia reconstruidos en información del dominio espacial. Para un fotograma predicho, el decodificador (450) combina residuos de predicción reconstruidos con predicciones con compensación de movimiento para formar un fotograma reconstruido. El decodificador (450) puede combinar de manera similar los residuos de predicción con las predicciones espaciales de la intrapredicción. Un bucle de compensación de movimiento en el decodificador de vídeo (450) incluye un filtro de desbloqueo adaptativo para suavizar las discontinuidades a través de las filas y/o columnas de límites de bloques en el fotograma decodificado (451).

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

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

#### V. Codificadores de vídeo de ejemplo.

La Figura 5 es un diagrama de bloques de un codificador de vídeo (500) 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 actual (505) y produce datos codificados (595) como salida.

El codificador (500) está basado en bloques y utiliza un formato de bloque que depende de la implementación. Los bloques pueden subdividirse aún más en diferentes etapas, *por ejemplo*, en las etapas de transformación de frecuencia y codificación de entropía. 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 subbloques de valores de píxeles para codificar y decodificar.

El sistema codificador (500) comprime los fotogramas predichos y los fotogramas intracodificados. A modo de presentación, la Figura 5 muestra una "intraruta" a través del codificador (500) para la codificación intrafotograma y una "interruta" para la codificación interfotograma. Muchos de los componentes del codificador

(500) se utilizan tanto para la codificación intrafotograma como para la codificación interfotograma. Las operaciones exactas realizadas por esos componentes pueden variar dependiendo del tipo de información que se esté comprimiendo.

5 Si el fotograma actual (505) es un fotograma predicho, un estimador de movimiento (510) estima el movimiento de bloques, subbloques u otros conjuntos de valores de píxeles del fotograma actual (505) con respecto a uno o más fotogramas de referencia (imágenes de referencia). El almacén de fotogramas (520) almacena en búfer uno o más fotogramas anteriores reconstruidos (525) para su uso como fotogramas de referencia (imágenes de referencia). Cuando se utilizan múltiples imágenes de referencia, las múltiples imágenes de referencia  
10 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 de movimiento (510) genera como información secundaria información de movimiento (515), tal como información de vector de movimiento diferencial, índices de referencia e información de modificación de la RPL. Durante la codificación, el codificador (500) construye las RPL de modo que se agregan nuevas imágenes de  
15 referencia cuando es apropiado, se eliminan imágenes de referencia más antiguas que ya no se utilizan para la compensación de movimiento cuando es apropiado y se reordenan las imágenes de referencia en las RPL cuando es apropiado.

En algunas implementaciones, al codificar un fotograma actual, el codificador (500) determina un RPS que  
20 incluye fotogramas de referencia en el almacén de fotogramas (520). El codificador (500) normalmente determina los RPS para el primer segmento del fotograma. En base a segmento por segmento, el codificador (500) crea una o más RPL para codificar un segmento determinado del fotograma actual. Para crear una RPL, el codificador (500) puede aplicar reglas sobre la selección de fotogramas de referencia disponibles en el RPS, en cuyo caso la información de modificación de la RPL no se señala explícitamente en los datos codificados  
25 (595). O, para crear la RPL, el codificador (500) puede seleccionar fotogramas de referencia específicos disponibles en el RPS, donde los fotogramas de referencia que se seleccionan se indicarán en la información de modificación de la RPL que se señala en los datos codificados (595). 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  
30 de modificación de la 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.

El compensador de movimiento (530) aplica vectores de movimiento reconstruidos al(a los) fotograma(s) de  
35 referencia reconstruido(s) (525) al formar un fotograma actual compensado por movimiento (535). La diferencia (si la hay) entre un subbloque, bloque, etc. del fotograma actual compensado por movimiento (535) y la parte correspondiente del fotograma actual original (505) es el residuo de predicción (545) para el subbloque, bloque, etc. Durante la reconstrucción posterior del fotograma actual, los residuos de predicción reconstruidos se agregan al fotograma actual compensado por movimiento (535) para obtener un fotograma reconstruido que  
40 sea más cercano al fotograma actual original (505). Sin embargo, en la compresión con pérdida, aún se pierde algo de información del fotograma actual original (505). La intraruta puede incluir un módulo de intrapredicción (no mostrado) que predice espacialmente los valores de píxeles de un bloque o subbloque actual a partir de valores de píxeles vecinos reconstruidos previamente.

Un transformador de frecuencia (560) convierte la información de vídeo del dominio espacial en información  
45 del dominio de frecuencia (*es decir*, datos espectrales, de transformación). Para fotogramas de vídeo basados en bloques, el transformador de frecuencia (560) aplica una transformada de coseno discreta, una aproximación entera de la misma u otro tipo de transformada de bloque hacia adelante a bloques o subbloques de datos de valores de píxeles o datos residuales de predicción, produciendo bloques/subbloques de coeficientes de  
50 transformación de frecuencia. Luego, un cuantizador (570) cuantiza los coeficientes de transformación. Por ejemplo, el cuantizador (570) aplica una cuantización escalar no uniforme a los datos del dominio de frecuencia con un tamaño de etapa que varía en base a fotograma por fotograma, en base a segmento por segmento, en base a bloque por bloque u otra base.

Cuando se necesita una versión reconstruida del fotograma actual para una estimación/compensación de  
55 movimiento subsiguiente, un cuantizador inverso (576) realiza una cuantización inversa en los datos del coeficiente de frecuencia cuantizados. Un transformador de frecuencia inversa (566) realiza una transformada de frecuencia inversa, produciendo bloques/subbloques de residuos de predicción reconstruidos o valores de píxeles. Para un fotograma predicho, el codificador (500) combina residuos de predicción reconstruidos (545)  
60 con predicciones con compensación por movimiento (535) para formar el fotograma reconstruido (505), que puede utilizarse como una imagen de referencia. (Aunque no se muestra en la Figura 5, en la intraruta, el codificador (500) puede combinar residuos de predicción con predicciones espaciales de la intrapredicción para reconstruir un fotograma que se utiliza como imagen de referencia). El almacén de fotogramas (520) almacena en búfer el fotograma actual reconstruido para utilizarlo como imagen de referencia en una predicción con  
65 compensación de movimiento subsiguiente.

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

El codificador de entropía (580) comprime la salida del cuantizador (570), así como la información de movimiento (515) y cierta información secundaria (por ejemplo, Valores QP, índices de referencia, información de modificación de la RPL). El codificador de entropía (580) proporciona datos codificados (595) al búfer (590), que multiplexa los datos codificados en un flujo de bits de salida. Los datos codificados (595) pueden incluir elementos de sintaxis que indican información de modificación de la RPL. La sección VII describe ejemplos de dichos elementos de sintaxis.

Un controlador (no mostrado) recibe entradas de diversos módulos del codificador. El controlador evalúa resultados intermedios durante la codificación, por ejemplo, estableciendo valores QP y realizando análisis de tasa-distorsión. El controlador trabaja con otros módulos para establecer y cambiar los parámetros de codificación durante la codificación. En particular, a la hora de decidir si se debe modificar y cómo hacerlo (por ejemplo, reemplazar, ajustar) las RPL, el controlador puede controlar qué imágenes de referencia se agregan a las RPL, controlar qué imágenes se eliminan de las RPL y reordenar las imágenes de referencia en las RPL para un direccionamiento más eficiente con índices de referencia. El controlador puede decidir eliminar imágenes de referencia de los RPS (y, por lo tanto, de las RPL), por ejemplo, eliminando todas las imágenes de referencia después de un cambio de escena, eliminando todas las imágenes de referencia después de la codificación de un tipo especial de imagen, como una imagen IDR, eliminando una imagen de referencia determinada después de que la utilización de la imagen de referencia para la compensación de movimiento caiga por debajo de una cantidad umbral y/o eliminando imágenes de referencia de acuerdo con otros criterios. El controlador puede decidir agregar imágenes de referencia al RPS, por ejemplo, agregando imágenes de acuerdo con el tipo de imagen/tipo de segmento en las imágenes, la capa temporal de las imágenes y/u otros criterios. Para una RPL, el controlador puede evaluar los resultados de la compensación de movimiento para los cuales no se modifica una RPL de acuerdo con elementos de sintaxis explícitamente señalados en el flujo de bits, y también evaluar los resultados de la compensación de movimiento para los cuales se modifica la RPL de acuerdo con elementos de sintaxis explícitamente señalados 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 la RPL) o seleccionar una RPL que haya sido modificada (como se especifica con la información de modificación de la RPL). Para 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 basado al menos en parte en la frecuencia de uso durante la codificación, y/o (c) agregar imágenes de referencia basado al menos en parte en la frecuencia de uso durante la codificación. Por ejemplo, el controlador puede decidir eliminar una imagen de referencia determinada 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 de acuerdo con otros criterios. O, el controlador puede decidir agregar una imagen de referencia determinada a la RPL si la utilización de la imagen de referencia para la compensación de movimiento supera una cantidad umbral y/o de acuerdo con otros criterios. O, el controlador puede decidir cómo reordenar las imágenes de referencia en la RPL basado en la frecuencia de utilización de las respectivas imágenes de referencia y/o de acuerdo con otros criterios. El controlador puede construir la(s) RPL en base a imagen por imagen, en base a segmento por segmento o sobre alguna otra base.

Dependiendo de la implementación y el tipo de compresión deseada, se pueden agregar módulos del codificador, omitirlos, dividirlos en múltiples módulos, combinarlos con otros módulos y/o reemplazarlos con módulos similares. En realizaciones alternativas, los codificadores con diferentes módulos y/u otras configuraciones de módulos realizan una o más de las técnicas descritas. Las realizaciones específicas de codificadores normalmente utilizan una variación o versión complementada del codificador (500). Las relaciones que se muestran entre los módulos dentro del codificador (500) indican flujos generales de información en el codificador; no se muestran otras relaciones por motivos de simplicidad.

#### VI. Decodificadores de vídeo de ejemplo.

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 codificados (695) para un fotograma comprimido o una secuencia de fotogramas y produce una salida que incluye un fotograma reconstruido (605), que puede utilizarse como una imagen de referencia. A modo de presentación, la Figura 6 muestra una "intraruta" a través del decodificador (600) para la decodificación intrafotograma y una "interruta" para la decodificación interfotograma. Muchos de los componentes del decodificador (600) se utilizan tanto para la decodificación intrafotograma como para la decodificación interfotograma. Las operaciones exactas

realizadas por esos componentes pueden variar dependiendo del tipo de información que se esté descomprimiendo.

Un búfer (690) recibe datos codificados (695) para un fotograma comprimido y hace que los datos codificados recibidos estén disponibles para el analizador/decodificador de entropía (680). Los datos codificados (695) pueden incluir elementos de sintaxis que indican información de modificación de la RPL. La sección VII describe ejemplos de dichos elementos de sintaxis. El analizador/decodificador de entropía (680) decodifica por entropía datos cuantizados codificados por entropía, así como información lateral codificada por entropía (incluidos índices de referencia, información de modificación de la RPL, etc.), aplicando normalmente la inversa de la codificación de entropía realizada en el codificador.

Durante la decodificación, el decodificador (600) construye las RPL de modo que se agregan nuevas imágenes de referencia cuando es apropiado, se eliminan imágenes de referencia más antiguas que ya no se utilizan para la compensación de movimiento cuando es apropiado y se reordenan las imágenes de referencia cuando es apropiado. El decodificador (600) puede construir las RPL basándose en la información disponible sobre las RPL (por ejemplo, imágenes de referencia disponibles en el RPS), modificaciones de acuerdo con reglas y/o de acuerdo con las modificaciones señaladas como parte de los datos codificados (695). En algunas implementaciones, por ejemplo, al decodificar un fotograma actual, el decodificador (600) determina un RPS que incluye fotogramas de referencia en el almacén de fotogramas (620). El decodificador (600) normalmente determina los RPS para el primer segmento del fotograma. En base a segmento por segmento, el decodificador (600) crea una o más RPL para decodificar un segmento determinado del fotograma actual. Para crear una RPL, en algunos casos (como se indica en los datos codificados (695)), el decodificador (600) aplica reglas sobre la selección de fotogramas de referencia disponibles en el RPS, en cuyo caso la información de modificación de la RPL no se analiza a partir de los datos codificados (695). En otros casos, para crear la RPL, el decodificador (600) selecciona fotogramas de referencia específicos disponibles en el RPS, donde los fotogramas de referencia que se seleccionan se indican en la información de modificación de la RPL que se analiza a partir de los datos codificados (695). 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 la 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.

Un compensador de movimiento (630) aplica información de movimiento (615) a una o más imágenes de referencia (625) para formar predicciones con compensación de movimiento (635) de subbloques y/o bloques del fotograma (605) que se está reconstruyendo. El almacén de fotogramas (620) almacena uno o más fotogramas previamente reconstruidos para utilizarlos como imágenes de referencia.

La intraruta puede incluir un módulo de intrapredicción (no mostrado) que predice espacialmente los valores de píxeles de un bloque o subbloque actual a partir de valores de píxeles vecinos reconstruidos previamente. En la interruta, el decodificador (600) reconstruye los residuos de predicción. Un cuantizador inverso (670) cuantiza inversamente los datos decodificados por entropía. Un transformador de frecuencia inversa (660) convierte los datos del dominio de frecuencia reconstruidos en información del dominio espacial. Por ejemplo, el transformador de frecuencia inversa (660) aplica una transformación de bloque inversa a los coeficientes de transformación de frecuencia, produciendo datos de valores de píxeles 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 de predicción reconstruidos (645) con predicciones con compensación de movimiento (635) para formar el fotograma reconstruido (605), que puede utilizarse como una imagen de referencia. (Aunque no se muestra en la Figura 6, en la intraruta, el decodificador (600) puede combinar residuos de predicción con predicciones espaciales de la intrapredicción para reconstruir un fotograma, que puede utilizarse como una imagen de referencia). Un bucle de compensación de movimiento en el decodificador (600) incluye un filtro de desbloqueo en bucle adaptativo (610) antes o después del almacén de fotogramas (620). El decodificador (600) aplica filtrado en bucle a los fotogramas reconstruidos para suavizar de forma adaptativa las discontinuidades en los límites de los fotogramas.

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

Dependiendo de la implementación y el tipo de descompresión deseada, se pueden agregar módulos del decodificador, omitirlos, dividirlos en múltiples módulos, combinarlos con otros módulos y/o reemplazarlos 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

decodificadores normalmente utilizan una variación o versión suplementada del decodificador (600). Las relaciones que se muestran entre los módulos dentro del decodificador (600) indican flujos generales de información en el decodificador; no se muestran otras relaciones por motivos de simplicidad.

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

En esta sección se presentan diversas innovaciones en el área de señalización de información de modificación de la RPL. En algunas situaciones, estas innovaciones dan como resultado una señalización más eficiente de los elementos de sintaxis para la información de modificación de la RPL.

10

### A. Imágenes de referencia y RPLs.

Una imagen de referencia es, en general, una imagen que contiene muestras que pueden utilizarse para la predicción interimagen en el proceso de decodificación de otras imágenes, que normalmente siguen a la imagen de referencia en el orden de decodificación. Es posible que haya múltiples imágenes de referencia disponibles en un momento determinado para su uso en la predicción con compensación de movimiento.

15

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 direccionan mediante 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 recientemente decodificadas, eliminar imágenes más antiguas que ya no se usan como imágenes de referencia y/o reordenar las imágenes de referencia dentro de la RPL para hacer que la señalización de los índices de referencia más comúnmente utilizados sea más eficiente. Un codificador y un decodificador pueden seguir las mismas reglas para construir, modificar, etc. sus RPL. Además de dichas reglas (o en lugar de dichas reglas), un codificador puede enviar información a un decodificador que indica cómo el decodificador debe construir, modificar, etc. sus RPL para que coincidan con las RPL utilizadas por el codificador. Normalmente, una RPL se construye durante la codificación y la decodificación basándose en la información disponible sobre la RPL (por ejemplo, imágenes disponibles en el RPS), modificaciones de acuerdo con las reglas y/o modificaciones señalizadas en el flujo de bits.

20

25

30

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 fotograma decodificado, como un búfer 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.

35

El codificador o decodificador determina los RPS una vez por imagen. Por ejemplo, el decodificador determina el RPS después de decodificar un encabezado de segmento para un segmento de la imagen, utilizando elementos de sintaxis señalados en el encabezado de segmento. Las imágenes de referencia se identifican con valores de recuento de orden de imagen ("POC"), partes de ellas y/u otra información señalizada en el flujo de bits. El codificador o decodificador determina grupos de imágenes de referencia de corto plazo e imágenes de referencia de largo plazo que pueden utilizarse en la predicción interimagen de la imagen actual (y que pueden utilizarse en la predicción interimagen de una o más de las imágenes que siguen a la imagen actual en el orden de decodificación). (El codificador o decodificador también determina grupos de imágenes de referencia que pueden utilizarse en la predicción interimagen 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). En conjunto, los grupos de imágenes de referencia son los RPS para la imagen actual.

40

45

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) combinando los grupos de imágenes de referencia de corto plazo e imágenes de referencia de largo plazo que pueden utilizarse en la predicción interimagen de la imagen actual. Para construir la RPL de acuerdo con las reglas de un enfoque "implícito", el codificador o decodificador puede utilizar las imágenes de referencia en la versión temporal de la RPL, o utilizar 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 la RPL no se señalará 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 utiliza información de modificación de la RPL señalizada en/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 imágenes de referencia en la RPL.

50

55

60

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

#### B. Señalización condicional de indicadores de modificación de la RPL.

De acuerdo con un aspecto de las innovaciones descritas en este documento, un codificador señala condicionalmente un indicador que indica si una RPL se modifica de acuerdo con elementos de sintaxis señalados explícitamente en el flujo de bits. Un decodificador correspondiente analiza condicionalmente dicho indicador.

En algunas implementaciones de ejemplo, el indicador es `ref_pic_list_modification_flag_10` o `ref_pic_list_modification_flag_l1` (generalmente, el indicador es `ref_pic_list_modification_flag_IX`, donde X puede ser 0 o 1). Si el valor del indicador `ref_pic_list_modification_flag_IX` es igual a 1, la RPL X se especifica explícitamente como una lista de valores `list_entry_IX[i]` (nuevamente, con X siendo 0 o 1). Si el valor del indicador `ref_pic_list_modification_flag_IX` es igual a 0, la RPL X se determina implícitamente. Cuando `ref_pic_list_modification_flag_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 `ref_pic_lists_modification()` en implementaciones de ejemplo. La estructura puede señalarse como parte de un encabezado de segmento. En la sintaxis de ejemplo (700), `ref_pic_list_modification_flag_IX` solo se envía cuando `NumPocTotalCurr` es mayor que 1. `NumPocTotalCurr` es una variable que indica un número total de imágenes de referencia aplicables para la codificación o decodificación actual. En las implementaciones de ejemplo de codificación o decodificación, cuando la variable `NumPocTotalCurr` se deriva para un segmento de una imagen actual, la variable indica el conteo de imágenes de referencia de corto plazo y de imágenes de referencia de largo plazo utilizadas como imágenes de referencia para codificar o decodificar la imagen actual.

Como se muestra en la Figura 7a, la señalización condicional de `ref_pic_list_modification_flag_IX` depende del valor de la variable `NumPocTotalCurr`. Cuando `NumPocTotalCurr` es menor o igual a 1, no hay posibilidad de modificación de la RPL y, por lo tanto, no es necesario enviar el indicador `ref_pic_list_modification_flag_IX`. Esta señalización condicional puede guardar uno o dos indicadores para cada segmento, cuando se cumple la condición. La modificación en la Figura 7a incluye la condición "si (`NumPocTotalCurr` > 1)" para determinar si se señala el indicador `ref_pic_list_modification_flag_IX`. 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).

Alternativamente, la señalización y el análisis de una estructura de modificación de la RPL que incluye uno o más indicadores de modificación de la RPL (por ejemplo, una estructura `ref_pic_lists_modification()`) se puede controlar evaluando una condición como parte del procesamiento del encabezado del segmento o de otra manera. La Figura 7b ilustra un enfoque para la señalización condicional y el análisis de los indicadores `ref_pic_list_modification_flag_10` y `ref_pic_list_modification_flag_l1` basado en esta condición. Específicamente, la Figura 7b muestra un ejemplo de sintaxis (750) para una estructura de sintaxis de encabezado de segmento que puede incluir una estructura de sintaxis `ref_pic_lists_modification()`, que se representa en la sintaxis (760) de la Figura 7c. Para la sintaxis de ejemplo (750) del encabezado del segmento, el indicador `lists_modification_present_flag` se señala en un conjunto de parámetros de imagen que se aplica a el segmento. Cuando `lists_modification_present_flag` es igual a 0, la estructura `ref_pic_lists_modification()` no está presente en el encabezado del segmento. Cuando `lists_modification_present_flag` es igual a 1, la estructura `ref_pic_lists_modification()` puede estar presente en el encabezado del segmento, dependiendo del valor de la variable `NumPocTotalCurr`. Si la variable `NumPocTotalCurr` es mayor que 1, entonces se señala la estructura `ref_pic_lists_modification()`, como se muestra en la sintaxis (760) de la Figura 7c. De lo contrario (la variable `NumPocTotalCurr` no es mayor que 1), no se señala la estructura `ref_pic_lists_modification()` y se infieren los valores de las entradas de la lista.

En las Figuras 7a-7c, 8 y 9, el término "u(n)" representa un entero sin signo que utiliza n bits. Cuando n es "v" (como en "u(v)"), el número de bits varía de manera dependiente del valor de otros elementos de sintaxis. El proceso de análisis 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 la lista.

De acuerdo con otro aspecto de las innovaciones descritas en este documento, un codificador señala condicionalmente elementos de sintaxis para entradas de la lista que indican cómo modificar una RPL. Un decodificador correspondiente analiza condicionalmente dichos elementos de sintaxis.

En algunas implementaciones de ejemplo, los elementos de sintaxis son para elementos de sintaxis `list_entry_10[i]` para la RPL 0 o elementos de sintaxis `list_entry_11[i]` para la RPL 1 (generalmente, el elemento de sintaxis es `list_entry_IX`, donde X puede ser 0 o 1). La Figura 8 muestra un ejemplo de sintaxis (800) para



una estructura de sintaxis `ref_pic_lists_modification()`, que puede señalarse como parte de un encabezado de segmento. En la sintaxis (800) de ejemplo, el elemento de sintaxis `list_entry_IX[0]` se señala condicionalmente en el flujo de bits. En particular, cuando `NumPocTotalCurr` es igual a 2 y `num_ref_idx_IX_active_minus1` es igual a 0, el elemento de sintaxis `list_entry_IX[0]` no se señala en el flujo de bits. La variable `num_ref_idx_IX_active_minus1` indica el índice de referencia máximo para la RPL X que se puede utilizar para decodificar un segmento. La variable `num_ref_idx_IX_active_minus1` puede tener un valor predeterminado (por ejemplo, un valor de 0... 15, como se especifica en el conjunto de parámetros de imagen aplicable), o `num_ref_idx_IX_active_minus1` puede tener un valor señalado en un encabezado de segmento para el segmento actual.

Como se muestra en la Figura 8, incluso cuando `ref_pic_list_modification_flag_IX` indica que la información de modificación de la RPL está señalizada en el flujo de bits, la señalización de `list_entry_IX[0]` depende de `NumPocTotalCurr` y `num_ref_idx_IX_active_minus1`. Cuando `NumPocTotalCurr` es igual a 2 y `num_ref_idx_IX_active_minus1` es igual a 1, el valor de `list_entry_IX[0]` se puede inferir basado en `ref_pic_list_modification_flag_IX`, ya que solo hay dos opciones posibles (valor predeterminado de 0 o valor no predeterminado de 1).

Por lo tanto, la Figura 8 incluye una condición que determina si se señalan los elementos de sintaxis de las entradas de la lista. Para RPL 0, la condición es "si ( `ref_pic_list_modification_flag_I0` && ( `NumPocTotalCurr` = 2 && `num_ref_idx_I0_active_minus1` = 0 ) )." Para RPL 1, la condición es "si ( `ref_pic_list_modification_flag_I1` && ! ( `NumPocTotalCurr` = 2 && `num_ref_idx_I1_active_minus1` = 0 ) )."

En el ejemplo de la Figura 8, `list_entry_I0[i]` especifica el índice de la imagen de referencia en `RefPicListTemp0` (una versión temporal de la RPL) que se colocará en la posición actual de la RPL 0. La longitud del elemento de sintaxis `list_entry_I0[i]` es  $\text{Ceil}(\log_2(\text{NumPocTotalCurr}))$  bits. El valor de `list_entry_I0[i]` está en el rango de 0 a `NumPocTotalCurr` - 1, inclusive. Si `NumPocTotalCurr` es igual a 2 y `num_ref_idx_I0_active_minus1` es igual a 0, se infiere que el elemento de sintaxis `list_entry_I0[0]` es igual a `ref_pic_list_modification_flag_I0`. De lo contrario, cuando el elemento de sintaxis `list_entry_I0[i]` no está presente, se infiere que es igual a 0.

En el ejemplo de la Figura 8, `list_entry_I1[i]` especifica el índice de la imagen de referencia en `RefPicListTemp1` (una versión temporal de la RPL) que se colocará en la posición actual de la RPL 1. La longitud del elemento de sintaxis `list_entry_I1[i]` es  $\text{Ceil}(\log_2(\text{NumPocTotalCurr}))$  bits. El valor de `list_entry_I1[i]` está en el rango de 0 a `NumPocTotalCurr` - 1, inclusive. Si `NumPocTotalCurr` es igual a 2 y `num_ref_idx_I1_active_minus1` es igual a 0, se infiere que el elemento de sintaxis `list_entry_I1[0]` es igual a `ref_pic_list_modification_flag_I1`. De lo contrario, cuando el elemento de sintaxis `list_entry_I1[i]` no está presente, se infiere que es igual a 0.

La Figura 9 muestra otro ejemplo de sintaxis (900) para una estructura de sintaxis `ref_pic_lists_modification()`, que puede señalarse como parte de un encabezado de segmento. En la sintaxis de ejemplo (900), el elemento de sintaxis `list_entry_IX[0]` se señala condicionalmente en el flujo de bits. Sin embargo, en comparación con la sintaxis de ejemplo de la Figura 8, la condición que se verifica es diferente. También, la señalización de elementos de sintaxis para `list_entry_IX[i]` se puede ajustar dependiendo de si se utiliza predicción ponderada.

De acuerdo con la Figura 9, el hecho de que la predicción ponderada esté habilitada o deshabilitada afecta la forma en que los elementos de sintaxis de las entradas de la lista se señalan en el flujo de bits. Para los segmentos P con `weighted_pred_flag` igual a 0 o para los segmentos B con `weighted_bipred_flag` igual a 0, la predicción ponderada está deshabilitada. De acuerdo con la sintaxis (900) de ejemplo de la Figura 9, cuando la predicción ponderada está deshabilitada, `list_entry_IX[0]` y `list_entry_IX[1]` no se envían cuando `NumPocTotalCurr` es igual a 2 y `num_ref_idx_IX_active_minus1` es igual a 1. En tal caso, se infiere que `list_entry_IX[0]` y `list_entry_IX[1]` son 1 y 0, respectivamente, ya que no habría sido necesaria la modificación de la RPL para la única otra posibilidad (es decir, que `list_entry_IX[0]` y `list_entry_IX[1]` sean iguales a 0 y 1, respectivamente).

Por lo tanto, la Figura 9 incluye una condición para si se señalan los elementos de sintaxis de las entradas de la lista. Para RPL 0, la condición es "si ( `ref_pic_list_modification_flag_I0` && ! ( `NumPocTotalCurr` = 2 && `num_ref_idx_I0_active_minus1` = 0 ) ) && ! ( `NumPocTotalCurr` = 2 && `num_ref_idx_I0_active_minus1` = 1 && ( ( `weighted_pred_flag` != 1 && `slice_type` = P ) | ( `weighted_bipred_flag` != 1 && `slice_type` = B ) ) )." Para RPL 1, la condición es "if ( `ref_pic_list_modification_flag_I1` && [ ( `NumPocTotalCurr` = 2 && `num_ref_idx_I1_active_minus1` = 0 ) && ! ( `NumPocTotalCurr` = 2 && `num_ref_idx_I1_active_minus1` = 1 && `weighted_bipred_flag` != 1 ) )."

Además, incluso en los casos en los que `NumPocTotalCurr` no es igual a 2 o `num_ref_idx_IX_active_minus1` no es igual a 1, cuando la predicción ponderada está deshabilitada (para segmentos P, `weighted_pred_flag` igual a 0; para segmentos B, `weighted_bipred_flag` igual a 0), la longitud del elemento de sintaxis `list_entry_IX[i]` está limitada a  $\text{Ceil}(\log_2(\text{NumPocTotalCurr} - i))$  bits. En este caso, sólo es útil colocar cada imagen de

referencia una vez en la lista, por lo tanto el número de posibilidades útiles disminuye a medida que aumenta el índice  $i$ .

En el ejemplo de la Figura 9,  $list\_entry\_l0[i]$  especifica el índice de la imagen de referencia en  $RefPicListTemp0$  (una versión temporal de la RPL) que se colocará en la posición actual de la RPL 0. Cuando la predicción ponderada está deshabilitada (para segmentos P,  $weighted\_pred\_flag$  igual a 0; para segmentos B,  $weighted\_bipred\_flag$  igual a 0), la longitud del elemento de sintaxis  $list\_entry\_l0[i]$  es  $Ceil(\log_2(\text{NumPocTotalCurr} - i))$  bits. De lo contrario, la longitud del elemento de sintaxis  $list\_entry\_l0[i]$  es  $Ceil(\log_2(\text{NumPocTotalCurr}))$  bits. Si  $\text{NumPocTotalCurr}$  es igual a 2 y  $num\_ref\_idx\_l0\_active\_minus1$  es igual a 0, se infiere que el elemento de sintaxis  $list\_entry\_l0[0]$  es igual a  $ref\_pic\_list\_modification\_flag\_l0$  (como en el ejemplo de la Figura 8). De lo contrario, si  $\text{NumPocTotalCurr}$  es igual a 2,  $num\_ref\_idx\_l0\_active\_minus1$  es igual a 1 y la predicción ponderada está deshabilitada (cuando  $weighted\_pred\_flag$  es igual a 0 y el segmento actual es un segmento P, o  $weighted\_bipred\_flag$  es igual a 0 y el segmento actual es un segmento B), se infiere que los elementos de sintaxis  $list\_entry\_l0[0]$  y  $list\_entry\_l0[1]$  son iguales a 1 y 0 respectivamente. De lo contrario, cuando el elemento de sintaxis  $list\_entry\_l0[i]$  no está presente, se infiere que es igual a 0.

Si la predicción ponderada está deshabilitada (cuando  $weighted\_pred\_flag$  es igual a 0 y el segmento actual es un segmento P, o  $weighted\_bipred\_flag$  es igual a 0 y el segmento actual es un segmento B), el valor de  $list\_entry\_l0[i]$  está en el rango de 0 a  $\text{NumPocTotalCurr} - (i + 1)$ , inclusive, y la lista  $RefPicListTemp0$  se acorta mediante la eliminación de cada entrada  $list\_entry\_l0[i]$  de la lista  $RefPicListTemp0$  después de que se analiza el valor de la entrada. De lo contrario, el valor de  $list\_entry\_l0[i]$  está en el rango de 0 a  $\text{NumPocTotalCurr} - 1$ , inclusive.

En el ejemplo de la Figura 9,  $list\_entry\_l1[i]$  especifica el índice de la imagen de referencia en  $RefPicListTemp1$  (una versión temporal de la RPL) que se colocará en la posición actual de la RPL 1. Si la predicción ponderada está deshabilitada ( $weighted\_bipred\_flag$  es igual a 0, ya que solo un segmento B utiliza la lista 1), la longitud del elemento de sintaxis  $list\_entry\_l1[i]$  es  $Ceil(\log_2(\text{NumPocTotalCurr} - i))$  bits. De lo contrario, la longitud del elemento de sintaxis  $list\_entry\_l1[i]$  es  $Ceil(\log_2(\text{NumPocTotalCurr}))$  bits. Si  $\text{NumPocTotalCurr}$  es igual a 2 y  $num\_ref\_idx\_l1\_active\_minus1$  es igual a 0, se infiere que el elemento de sintaxis  $list\_entry\_l1[0]$  es igual a  $ref\_pic\_list\_modification\_flag\_l1$  (como en el ejemplo de la Figura 8). De lo contrario, si  $\text{NumPocTotalCurr}$  es igual a 2,  $num\_ref\_idx\_l1\_active\_minus1$  es igual a 1 y la predicción ponderada está deshabilitada ( $weighted\_bipred\_flag$  es igual a 0 - el segmento actual es un segmento B), los elementos de sintaxis  $list\_entry\_l1[0]$  y  $list\_entry\_l1[1]$  se infieren como iguales a 1 y 0 respectivamente. De lo contrario, cuando el elemento de sintaxis  $list\_entry\_l1[i]$  no está presente, se infiere que es igual a 0.

Si la predicción ponderada está deshabilitada ( $weighted\_bipred\_flag$  es igual a 0 - el segmento actual es un segmento B), el valor de  $list\_entry\_l1[i]$  está en el rango de 0 a  $\text{NumPocTotalCurr} - (i + 1)$ , inclusive, y la lista  $RefPicListTemp1$  se acorta mediante la eliminación de cada entrada  $list\_entry\_l1[i]$  de la lista  $RefPicListTemp1$  después de que se analiza el valor de la entrada. De lo contrario, el valor de  $list\_entry\_l0[i]$  está en el rango de 0 a  $\text{NumPocTotalCurr} - 1$ , inclusive.

D. Técnicas generalizadas para la señalización condicional y el análisis de indicadores de modificación de la RPL.

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

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 de ejemplo, la variable es  $\text{NumPocTotalCurr}$  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 único factor (por ejemplo, el valor de una variable que indica un número total de imágenes de referencia), o la condición que se evalúa (1010) puede incluir múltiples factores (por ejemplo, el valor de una variable que indica un número total de imágenes de referencia, así como uno o más factores adicionales). La condición se puede evaluar (1010) como parte del procesamiento de una estructura de modificación de la RPL. O, la condición se puede evaluar (1010) como parte del procesamiento de un encabezado de segmento.

Dependiendo de los resultados de la evaluación, el dispositivo señala condicionalmente (1020) en un flujo de bits un indicador que indica si una RPL se modifica (por ejemplo, se reemplaza, se ajusta) de acuerdo con elementos de sintaxis señalados explícitamente en el flujo de bits. Por ejemplo, el indicador es uno de  $ref\_pic\_list\_modification\_flag\_l0$  o  $ref\_pic\_list\_modification\_flag\_l1$ , y puede ser señalizada condicionalmente como parte de una estructura de modificación de la RPL de un encabezado de segmento. O, después de que se evalúa la condición (1010), dependiendo de los resultados de la evaluación, la estructura de modificación

de la RPL (incluyendo uno o más indicadores que indican si una RPL se modifica de acuerdo con elementos de sintaxis señalados explícitamente en el flujo de bits) se señala condicionalmente en el flujo de bits.

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

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

10 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 de ejemplo, la variable es NumPocTotalCurr 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  
15 único factor (por ejemplo, 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 adicionales). La condición se puede evaluar (1110) como parte del procesamiento de una estructura de modificación de la RPL. O, la condición se puede evaluar (1110) como parte del procesamiento de un encabezado de segmento.

20 Dependiendo de los resultados de la evaluación, el dispositivo analiza condicionalmente (1120) a partir de un flujo de bits un indicador que indica si se modifica una RPL (por ejemplo, reemplazado, ajustado) de acuerdo con los elementos de sintaxis señalados explícitamente en el flujo de bits. Por ejemplo, el indicador es uno de ref\_pic\_list\_modification\_flag\_10 o ref\_pic\_list\_modification\_flag\_11, y puede ser señalizada condicionalmente  
25 como parte de una estructura de modificación de la RPL de un encabezado de segmento. O, después de evaluar la condición (1110), dependiendo de los resultados de la evaluación, la estructura de modificación de la RPL (que incluye uno o más indicadores que indican si una RPL se modifica de acuerdo con elementos de sintaxis señalados explícitamente en el flujo de bits) se analiza condicionalmente a partir del flujo de bits.

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

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

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

40 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, NumPocTotalCurr en algunas implementaciones de ejemplo). O, la condición depende, al menos en parte, de un número de imágenes de referencia activas para la RPL. O, la condición depende, al menos en parte, de si la predicción ponderada está deshabilitada. Se puede utilizar una lógica diferente para verificar si la predicción ponderada está deshabilitada  
45 dependiendo de si un segmento actual es un segmento P o un segmento B y/o dependiendo de qué RPL se está señalando/analizando. Por ejemplo, la lógica para verificar la condición de una primera RPL (que puede 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.

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

55 En algunas implementaciones de ejemplo, 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 el(los) elemento(s) 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 de ejemplo, 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)  
60 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.

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

65

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

- 5 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, NumPocTotalCurr en algunas implementaciones de ejemplo). O, la condición depende, al menos en parte, de un número de imágenes de referencia activas para la RPL. O, la condición depende, al menos en parte, de si la predicción ponderada está deshabilitada. Se puede utilizar 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 está señalando/analizando. Alternativamente, el decodificador evalúa otras y/o condiciones adicionales.

10 Dependiendo de los resultados de la evaluación, el dispositivo analiza condicionalmente (1320) a partir 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 de las entradas de la lista se analizan condicionalmente a partir de una estructura de modificación de la RPL de un encabezado de segmento.

20 En algunas implementaciones de ejemplo, 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 el(los) elemento(s) 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 de ejemplo, 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 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.

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

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

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

35 El dispositivo evalúa (1410) una condición. Por ejemplo, la condición depende al menos en parte de si la predicción ponderada está deshabilitada. Se puede utilizar 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 está señalando/analizando. Por ejemplo, la lógica para verificar la condición de una primera RPL (que puede 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 utilizado por un segmento B). Alternativamente, el codificador evalúa otras y/o condiciones adicionales.

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

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

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

60 El decodificador evalúa (1510) una condición. Por ejemplo, la condición depende al menos en parte de si la predicción ponderada está deshabilitada. Se puede utilizar 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 está señalando/analizando. Alternativamente, el decodificador evalúa otras y/o condiciones adicionales.

Dependiendo de los resultados de la evaluación, el dispositivo ajusta (1520) el análisis 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 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 de ejemplo, si la predicción ponderada está deshabilitada, la longitud de un elemento de sintaxis determinado para la entrada de la lista[ $i$ ] es  $\text{Ceil}(\text{Log2}(\text{NumPocTotalCurr}-i))$ . De lo contrario (la predicción ponderada está habilitada), la longitud del elemento de sintaxis determinado para la entrada de la lista[ $i$ ] es  $\text{Ceil}(\text{Log2}(\text{NumPocTotalCurr}))$  bits.

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

#### G. Alternativas.

Las Figuras 7a, 7b, 10 y 11 ilustran la señalización condicional y el análisis de un indicador como `ref_pic_list_modification_flag_10` o `ref_pic_list_modification_flag_l1` basado en una condición. De esta manera, se controla la señalización de información de modificación de la RPL adicional (como elementos de sintaxis para entradas de la lista). Como se explica con referencia a la Figura 7a, la señalización y el análisis de un indicador de modificación de la RPL se pueden controlar evaluando la condición como parte de una estructura `ref_pic_lists_modification()`. Alternativamente, como se explica con referencia a la Figura 7b, la señalización y el análisis de la estructura de modificación de la RPL (por ejemplo, la estructura `ref_pic_lists_modification()`) se puede controlar evaluando la misma condición como parte del procesamiento del encabezado del segmento o de otra manera. Por ejemplo, si la variable `NumPocTotalCurr` es mayor que 1, entonces se señala la estructura `ref_pic_lists_modification()`. De lo contrario (la variable `NumPocTotalCurr` no es mayor que 1), la estructura `ref_pic_lists_modification()` no se señala y los valores de las entradas de la lista se infieren como se describe anteriormente con referencia a la Figura 7a. Extendiendo la Figura 10, después de evaluar la condición, dependiendo de los resultados de la evaluación, la estructura de sintaxis de modificación de la RPL se señala condicionalmente. Extendiendo la Figura 11, después de evaluar la condición, dependiendo de los resultados de la evaluación, la estructura de sintaxis de modificación de la RPL se analiza condicionalmente.

A modo de ilustración, la descripción detallada incluye diversos ejemplos con nombres específicos para algunos parámetros y variables. Las innovaciones descritas en este documento no están limitadas a implementaciones con parámetros o variables que tengan dichos nombres. En cambio, las innovaciones descritas en este documento pueden implementarse con diversos tipos de parámetros y variables.

# REIVINDICACIONES

1. Un método en un dispositivo informático que implementa un decodificador de vídeo (450; 600), el método que comprende:  
5 recibir datos codificados en un flujo de bits; y  
decodificar los datos codificados para producir fotogramas de vídeo reconstruidos, incluyendo:  
10 como parte del procesamiento de un encabezado de segmento de un segmento, evaluar (1100) si un valor de una variable es mayor que 1, indicando la variable un número total de imágenes de referencia; y  
solo cuando el valor es mayor que 1, analizar condicionalmente (1120) desde el flujo de bits:  
15 (a) un primer indicador que indica si una primera lista de imágenes de referencia, RPL, se modifica de acuerdo con elementos de sintaxis señalados explícitamente en el flujo de bits; y  
(b) si el tipo de segmento del segmento es B, un segundo indicador que indica si se modifica una segunda RPL de acuerdo con los elementos de sintaxis señalados explícitamente en el flujo de bits.  
20 2. El método de la reivindicación 1, en el que el primer indicador y, si el tipo de segmento del segmento es B, el segundo indicador se analizan condicionalmente a partir de una estructura de modificación de la RPL.  
25 3. El método de la reivindicación 1 o 2, en el que una estructura de modificación de la RPL que incluye el primer indicador y, si el tipo de segmento del segmento es B, el segundo indicador se analiza condicionalmente dependiendo de si se cumple la condición.  
30 4. El método de una cualquiera de las reivindicaciones 1 a 3, en el que una imagen actual incluye el segmento, el número de imágenes de referencia totales considerando las imágenes de referencia a corto plazo utilizadas en la decodificación de la imagen actual y considerando las imágenes de referencia a largo plazo utilizadas en la decodificación para la imagen actual.  
35 5. El método de una cualquiera de las reivindicaciones 1 a 4, en el que los elementos de sintaxis señalados explícitamente en el flujo de bits especifican múltiples entradas de la lista de la primera RPL y, si el tipo de segmento del segmento es B, múltiples entradas de la lista de la segunda RPL, en el que cada una de las entradas de la lista especifica un índice de una imagen de referencia en una versión temporal de la primera RPL que se colocará en una posición actual de la primera RPL.  
40 6. El método de una cualquiera de las reivindicaciones 1 a 5, en el que la decodificación de los datos codificados incluye además analizar a partir del flujo de bits otro indicador, el otro indicador que indica si una estructura de modificación de la RPL puede estar presente en el flujo de bits, y en el que el otro indicador se señala como parte de un conjunto de parámetros de imagen.  
45 7. Un sistema informático que implementa un sistema codificador de vídeo (450; 600) que comprende:  
un búfer configurado para recibir fotogramas de vídeo; y  
un codificador de vídeo configurado para realizar la codificación de los fotogramas de vídeo para producir datos  
50 codificados en un flujo de bits, en donde la codificación de los fotogramas de vídeo incluye:  
evaluar los resultados de la compensación de movimiento para la cual se modifica una lista de imágenes de referencia, RPL, de acuerdo con elementos de sintaxis señalados explícitamente en el flujo de bits;  
55 evaluar los resultados de la compensación de movimiento para los cuales la RPL no se modifica de acuerdo con los elementos de sintaxis señalados explícitamente en el flujo de bits;  
modificar la RPL para lograr uno o más de (a) reordenar una o más imágenes de referencia para un direccionamiento más eficiente con índices de referencia, (b) eliminar una o más imágenes de referencia  
60 basadas al menos en parte en la frecuencia de uso durante la codificación, y (c) agregar una o más imágenes de referencia basadas al menos en parte en la frecuencia de uso durante la codificación;  
evaluar si un valor de una variable es mayor que 1, indicando la variable un número total de imágenes de  
65 referencia; y

sólo cuando el valor es mayor que 1, señalar condicionalmente en el flujo de bits un indicador que indica si la RPL se modifica de acuerdo con dichos elementos de sintaxis señalizados explícitamente en el flujo de bits.

5 8. El sistema informático de la reivindicación 7, en el que el indicador se señala condicionalmente como parte de una estructura de modificación de la RPL, y/o

10 en el que el indicador es un primer indicador que indica si una primera RPL se modifica de acuerdo con elementos de sintaxis señalizados explícitamente en el flujo de bits, y en el que la estructura de modificación de la RPL incluye además un segundo indicador que indica si una segunda RPL se modifica de acuerdo con elementos de sintaxis señalizados explícitamente en el flujo de bits.

15 9. El sistema informático de la reivindicación 7 u 8, en el que la condición se evalúa como parte del procesamiento de un encabezado de segmento de un segmento, y/o

en la que una imagen actual incluye el segmento, el número de imágenes de referencia totales considerando las imágenes de referencia a corto plazo utilizadas en la codificación de la imagen actual y considerando las imágenes de referencia a largo plazo utilizadas en la codificación para la imagen actual.

20 10. El sistema informático de una cualquiera de las reivindicaciones 7 a 9, en la que una estructura de modificación de la RPL que incluye el indicador se señala condicionalmente dependiendo de si se cumple la condición.

25 11. El sistema informático de una cualquiera de las reivindicaciones 7 a 10, en el que los elementos de sintaxis señalizados explícitamente en el flujo de bits especifican múltiples entradas de la lista de la RPL.

30 12. El sistema informático de una cualquiera de las reivindicaciones 7 a 11, en el que la codificación de los fotogramas de vídeo incluye además la señalización en el flujo de bits de otro indicador, el otro indicador que indica si una estructura de modificación de la RPL puede estar presente en el flujo de bits, y en el que el otro indicador se señala como parte de un conjunto de parámetros de imagen.

13. Uno o más medios legibles por ordenador que tienen almacenadas en ellos instrucciones ejecutables por ordenador para ocasionar que una unidad de procesamiento realice un método que comprende:

35 recibir datos codificados en un flujo de bits; y

decodificar los datos codificados para producir fotogramas de vídeo reconstruidos, incluyendo:

40 como parte del procesamiento de un encabezado de segmento de un segmento, evaluar (1100) si un valor de una variable es mayor que 1, la variable que indica un número total de imágenes de referencia; y

sólo cuando el valor es mayor que 1, analizar condicionalmente (1120) desde el flujo de bits:

45 (a) un primer indicador que indica si una primera lista de imágenes de referencia ("RPL") se modifica de acuerdo con elementos de sintaxis señalizados explícitamente en el flujo de bits; y

(b) si el tipo de segmento del segmento es B, un segundo indicador que indica si se modifica una segunda RPL de acuerdo con los elementos de sintaxis señalizados explícitamente en el flujo de bits.

50 14. El uno o más medios legibles por ordenador de la reivindicación 13, en el que el primer indicador y, si el tipo de segmento del segmento es B, el segundo indicador se analiza condicionalmente a partir de una estructura de modificación de la RPL, y/o

55 en la que una estructura de modificación de la RPL que incluye el primer indicador y, si el tipo de segmento del segmento es B, el segundo indicador se analiza condicionalmente dependiendo de si se cumple la condición.

15. El uno o más medios legibles por ordenador de la reivindicación 13 o 14, en la que al menos una de:

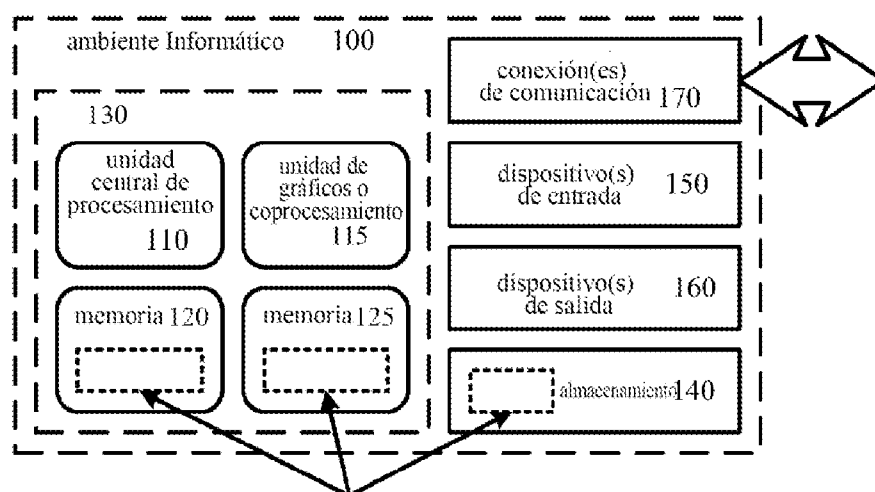
60 una imagen actual incluye el segmento, el número de imágenes de referencia totales considerando las imágenes de referencia a corto plazo utilizadas en la decodificación de la imagen actual y considerando las imágenes de referencia a largo plazo utilizadas en la decodificación para la imagen actual;

los elementos de sintaxis señalizados explícitamente en el flujo de bits especifican múltiples entradas de la lista de la primera RPL y, si el tipo de segmento del segmento es B, múltiples entradas de la lista de la segunda RPL;

65

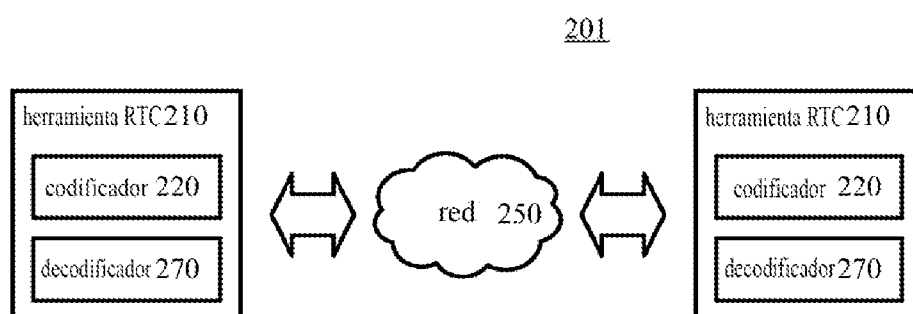
la decodificación de los datos codificados incluye además analizar a partir del flujo de bits de otro indicador, el otro indicador que indica si puede estar presente una estructura de modificación de la RPL en el flujo de bits, y en el que el otro indicador se señala como parte de un conjunto de parámetros de imagen.



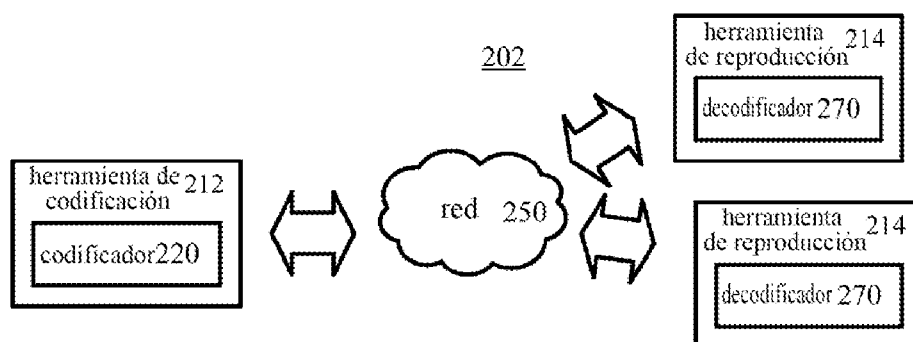


software 180 que implementa una o más innovaciones para la señalización de información de modificación de una lista de imágenes de referencia

**Figura 1**



**Figura 2a**



**Figura 2b**

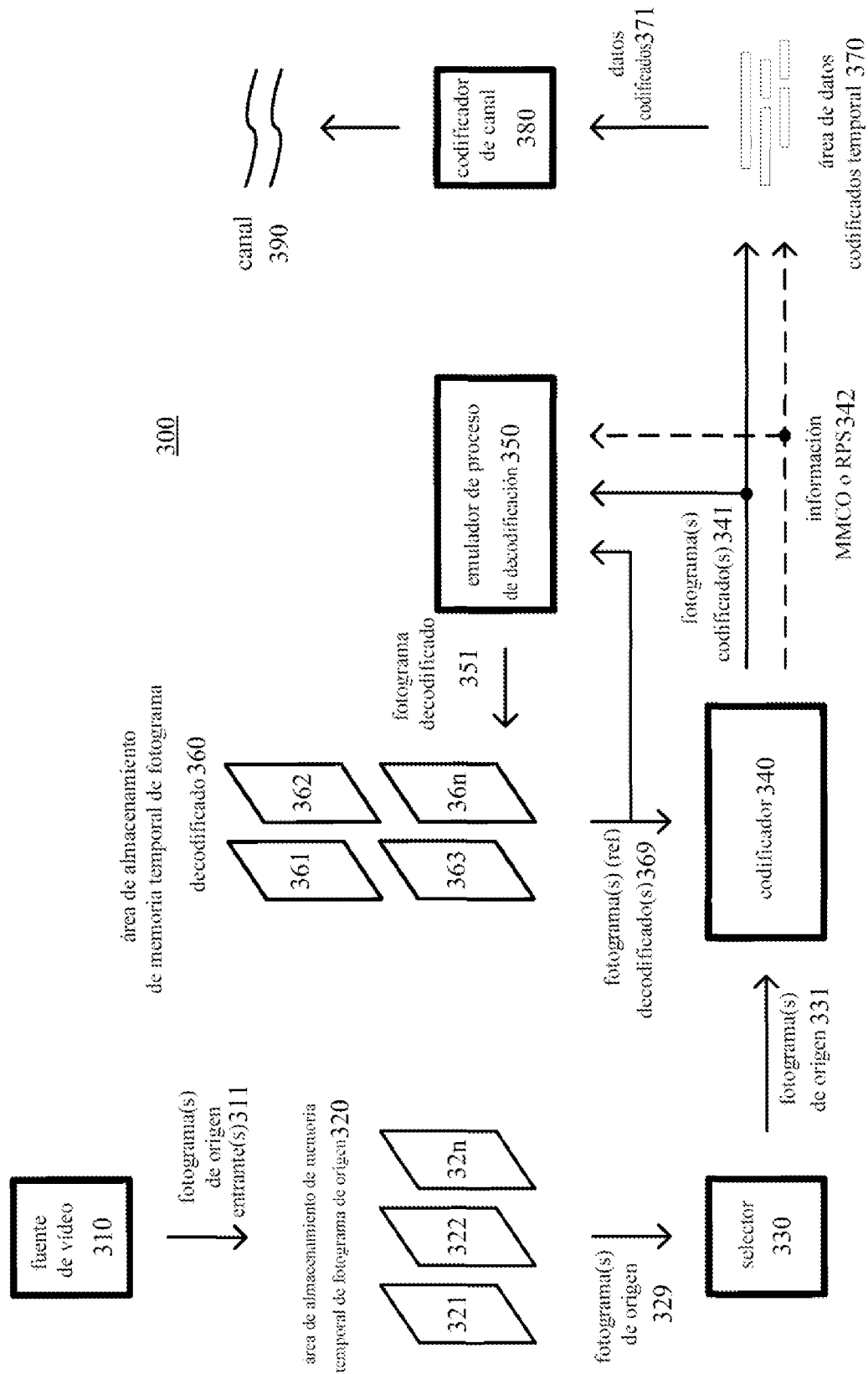
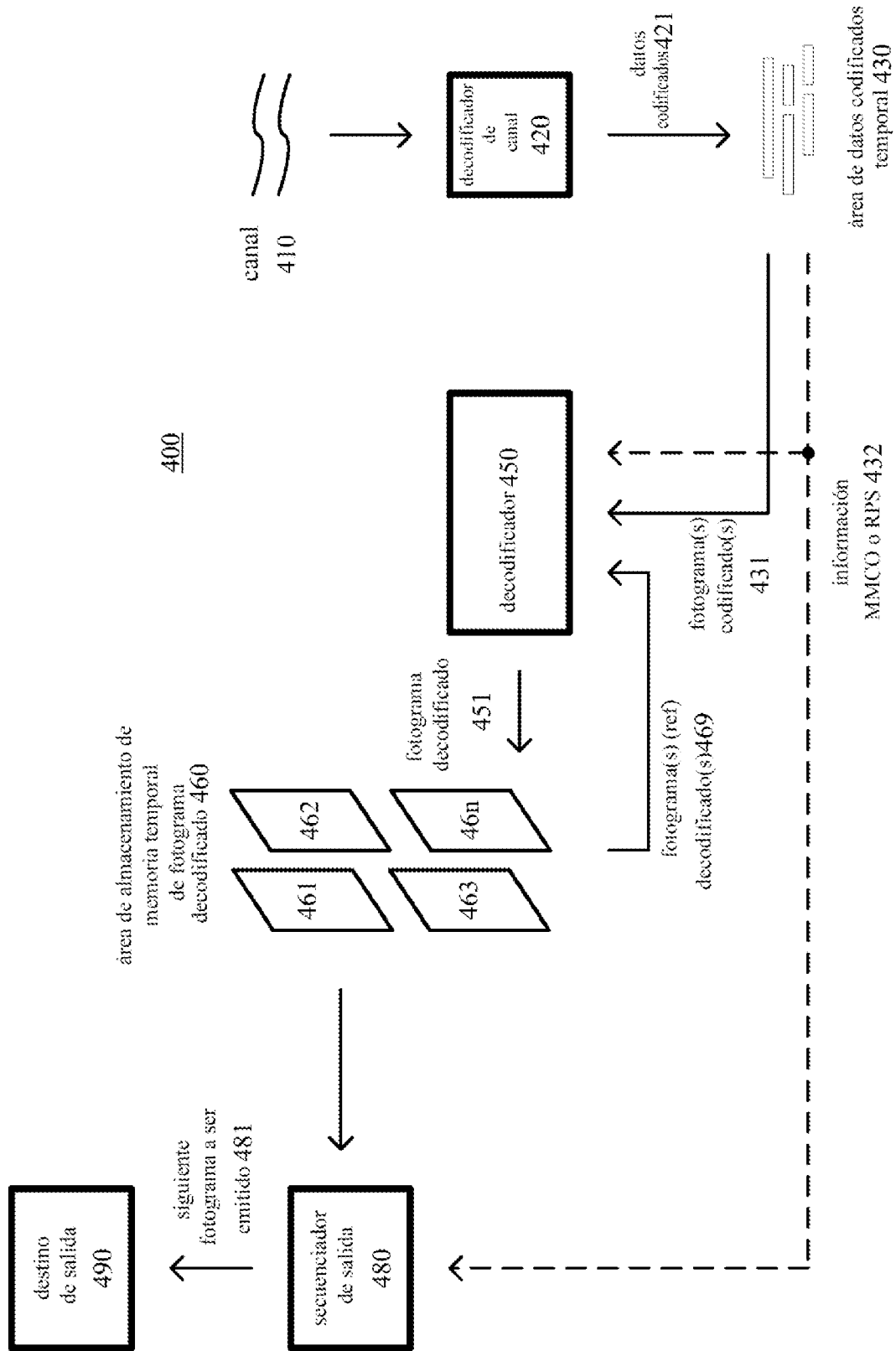
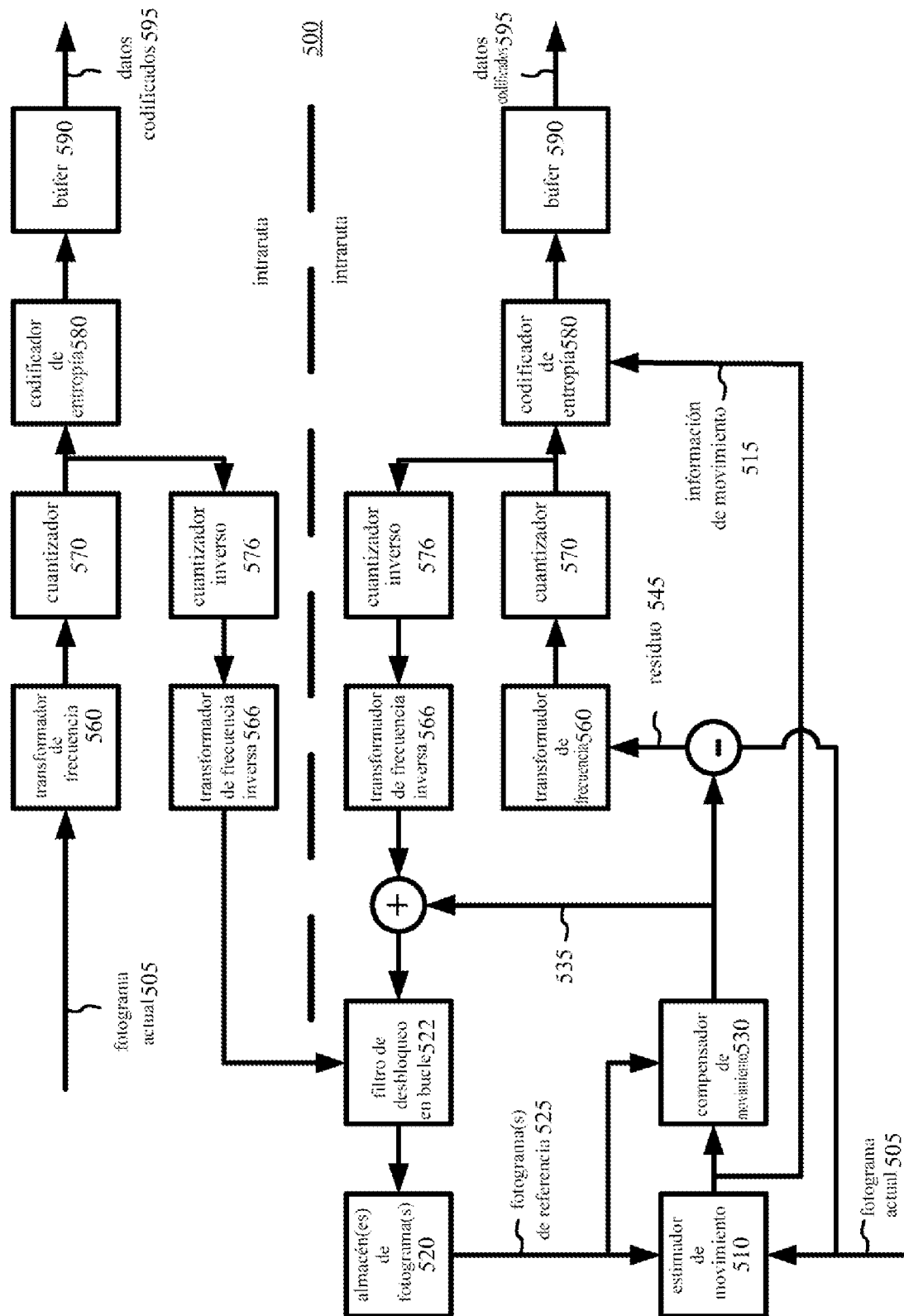


Figura 3



**Figura 4**



**Figura 5**

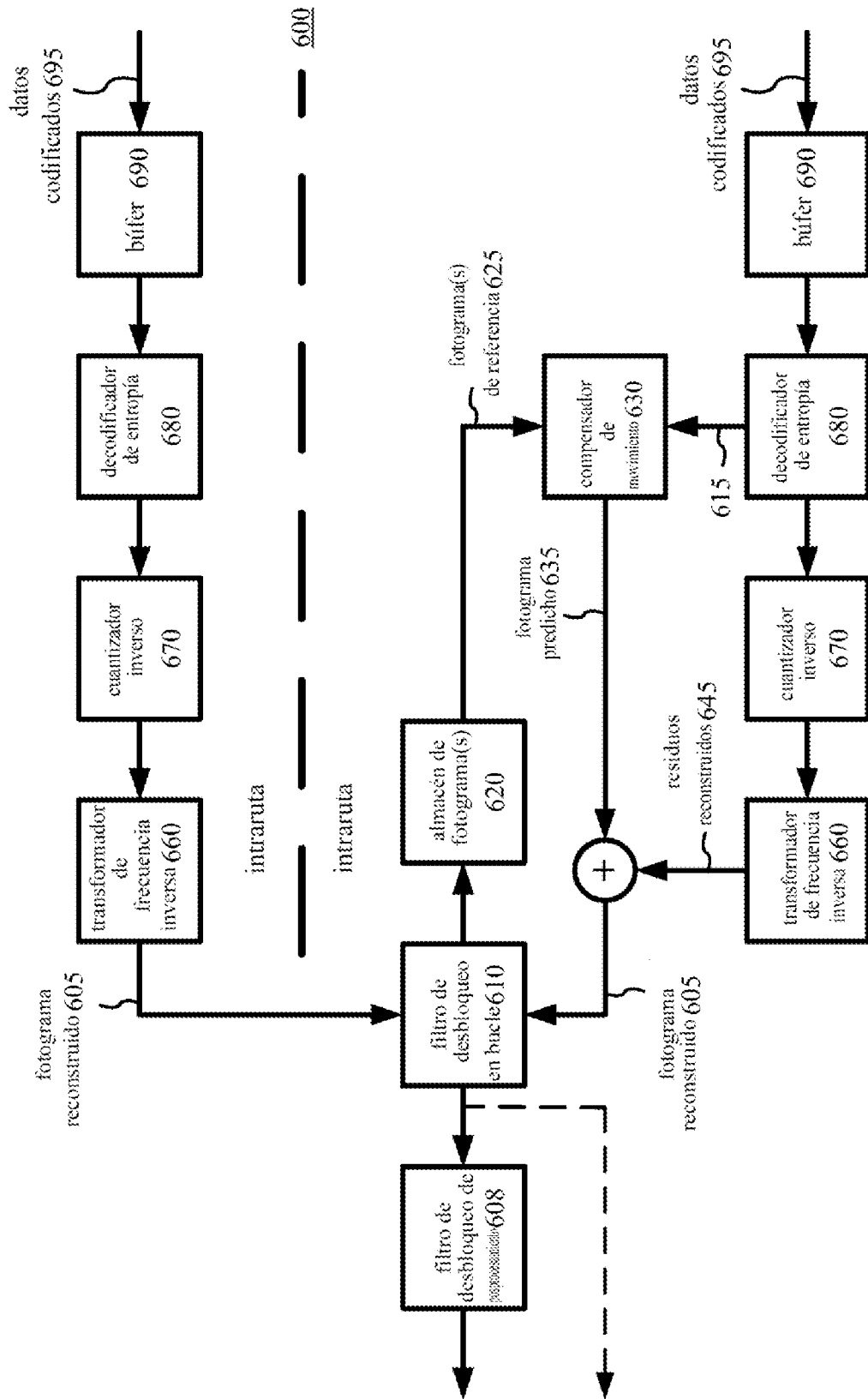


Figura 6

700

ref_pic_lists_modification( ) {	<b>Descriptor</b>
if( NumPocTotalCurr > 1 )	
<b>ref_pic_list_modification_flag_l0</b>	u(1)
if( ref_pic_list_modification_flag_l0 )	
for( i = 0; i <= num_ref_idx_l0_active_minus1; i++ )	
<b>list_entry_l0[ i ]</b>	u(v)
if( slice_type == B ) {	
if( NumPocTotalCurr > 1 )	
<b>ref_pic_list_modification_flag_l1</b>	u(1)
if( ref_pic_list_modification_flag_l1 )	
for( i = 0; i <= num_ref_idx_l1_active_minus1; i++ )	
<b>list_entry_l1[ i ]</b>	u(v)
}	
}	

**Figura 7a**

750

slice_segment_header( ) {	Descriptor
...	
if( slice_type == P    slice_type == B ) {	
<b>num_ref_idx_active_override_flag</b>	u(1)
if( num_ref_idx_active_override_flag ) {	
<b>num_ref_idx_l0_active_minus1</b>	ue(v)
if( slice_type == B )	
<b>num_ref_idx_l1_active_minus1</b>	ue(v)
}	
if( list_modification_present_flag && NumPocTotalCurr > 1 )	
ref_pic_lists_modification( )	
...	
}	

Figura 7b

760

ref_pic_lists_modification( ) {	Descriptor
<b>ref_pic_list_modification_flag_l0</b>	u(1)
if( ref_pic_list_modification_flag_l0 )	
for( i = 0; i <= num_ref_idx_l0_active_minus1; i++ )	
<b>list_entry_l0[ i ]</b>	u(v)
if( slice_type == B ) {	
<b>ref_pic_list_modification_flag_l1</b>	u(1)
if( ref_pic_list_modification_flag_l1 )	
for( i = 0; i <= num_ref_idx_l1_active_minus1; i++ )	
<b>list_entry_l1[ i ]</b>	u(v)
}	
}	

Figura 7c

800

ref_pic_lists_modification() {	Descriptor
<b>ref_pic_list_modification_flag_l0</b>	u(1)
if( ref_pic_list_modification_flag_l0 && !( NumPocTotalCurr == 2 && num_ref_idx_l0_active_minus1 == 0 ) )	
for( i = 0; i <= num_ref_idx_l0_active_minus1; i++ )	
<b>list_entry_l0[ i ]</b>	u(v)
if( slice_type == B ) {	
<b>ref_pic_list_modification_flag_l1</b>	u(1)
if( ref_pic_list_modification_flag_l1 && !( NumPocTotalCurr == 2 && num_ref_idx_l1_active_minus1 == 0 ) )	
for( i = 0; i <= num_ref_idx_l1_active_minus1; i++ )	
<b>list_entry_l1[ i ]</b>	u(v)
}	
}	

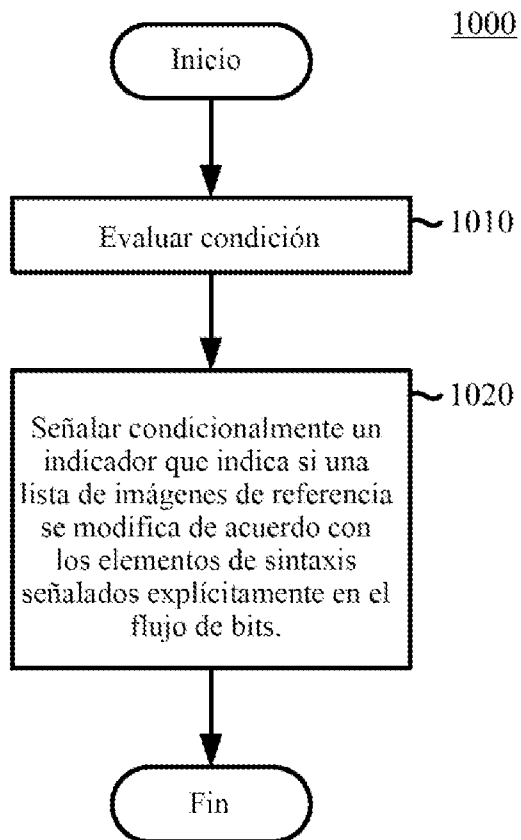
Figura 8

900

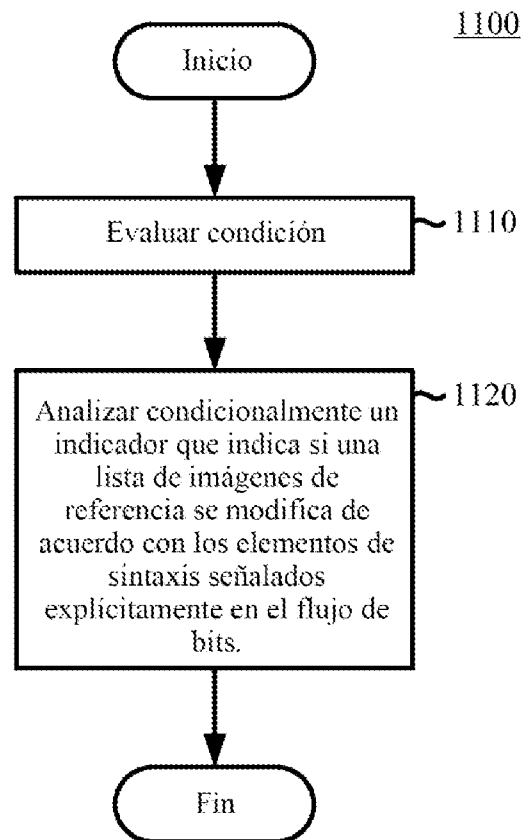
ref_pic_lists_modification() {	Descriptor
<b>ref_pic_list_modification_flag_l0</b>	u(1)
if( ref_pic_list_modification_flag_l0 && !( NumPocTotalCurr == 2 && num_ref_idx_l0_active_minus1 == 0 ) && !( NumPocTotalCurr == 2 && num_ref_idx_l0_active_minus1 == 1 && ( ( weighted_pred_flag != 1 && slice_type == P )    ( weighted_bipred_flag != 1 && slice_type == B ) ) ) )	
for( i = 0; i <= num_ref_idx_l0_active_minus1; i++ )	
<b>list_entry_l0[ i ]</b>	u(v)
if( slice_type == B ) {	
<b>ref_pic_list_modification_flag_l1</b>	u(1)
if( ref_pic_list_modification_flag_l1 && !( NumPocTotalCurr == 2 && num_ref_idx_l1_active_minus1 == 0 ) && !( NumPocTotalCurr == 2 && num_ref_idx_l1_active_minus1 == 1 && weighted_bipred_flag != 1 ) )	
for( i = 0; i <= num_ref_idx_l1_active_minus1; i++ )	
<b>list_entry_l1[ i ]</b>	u(v)
}	
}	

Figura 9

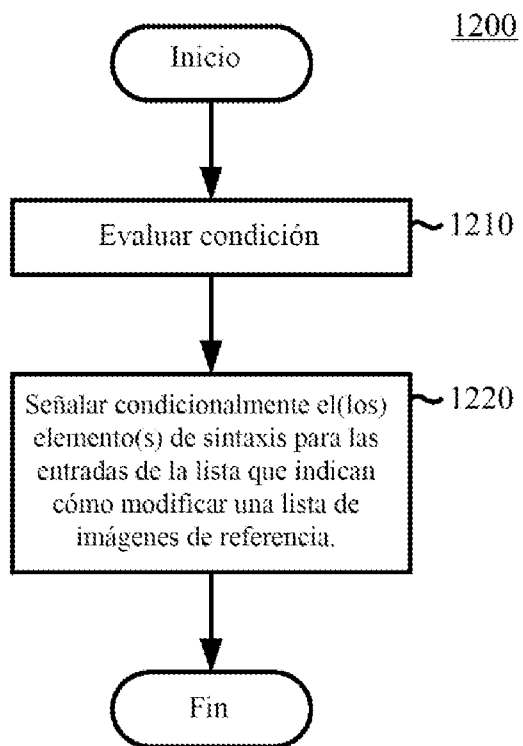




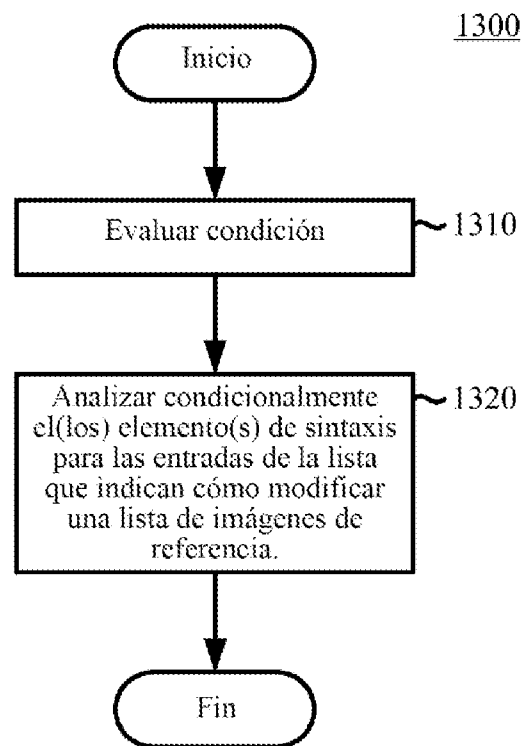
**Figura 10**



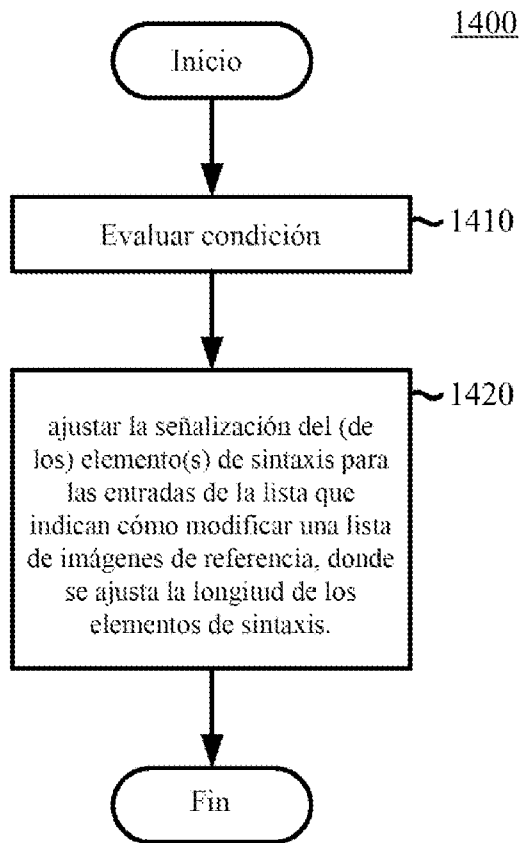
**Figura 11**



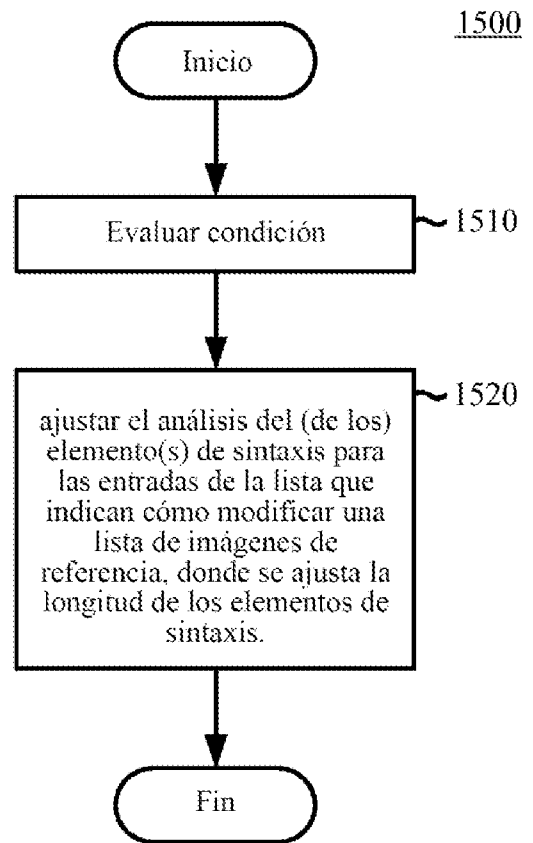
**Figura 12**



**Figura 13**



**Figura 14**



**Figura 15**