

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 742 027**

51 Int. Cl.:

**H04N 19/70** (2014.01)

**H04N 19/13** (2014.01)

**H04N 19/30** (2014.01)

**H04N 19/463** (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **02.04.2013 PCT/US2013/034968**

87 Fecha y número de publicación internacional: **17.10.2013 WO13154866**

96 Fecha de presentación y número de la solicitud europea: **02.04.2013 E 13717116 (1)**

97 Fecha y número de publicación de la concesión europea: **15.05.2019 EP 2837178**

54 Título: **Omitir binarios para la codificación de índices de referencia en la codificación de vídeo**

30 Prioridad:

**11.04.2012 US 201261623043 P**

**23.04.2012 US 201261637218 P**

**30.04.2012 US 201261640568 P**

**15.05.2012 US 201261647422 P**

**27.06.2012 US 201261665151 P**

**14.03.2013 US 201313828173**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**12.02.2020**

73 Titular/es:

**QUALCOMM INCORPORATED (100.0%)  
International IP Administration, 5775 Morehouse  
Drive**

**San Diego, California 92121-1714 , US**

72 Inventor/es:

**KARCZEWICZ, MARTA;  
SEREGIN, VADIM;  
WANG, XIANGLIN y  
COBAN, MUHAMMED ZEYD**

74 Agente/Representante:

**FORTEA LAGUNA, Juan José**

ES 2 742 027 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

Omitir binarios para la codificación de índices de referencia en la codificación de vídeo

5 **[0001]** Esta divulgación reclama el beneficio de la Solicitud Provisional Estadounidense N° 61/623.043, presentada el 11 de abril de 2012, la Solicitud Provisional Estadounidense 61/637.218, presentada el 23 de abril de 2012, la Solicitud Provisional Estadounidense 61/640.568, presentada el 30 de abril de 2012, la Solicitud Provisional Estadounidense 61/647.422, presentada el 15 de mayo de 2012, y la Solicitud Provisional Estadounidense N° 61/665.151, presentada el 27 de junio de 2012.

10

## CAMPO TÉCNICO

**[0002]** La presente divulgación se refiere a la codificación de vídeo y, de forma más particular, a técnicas para codificar elementos sintácticos en un proceso de codificación de vídeo.

15

## ANTECEDENTES

**[0003]** Las capacidades del vídeo digital pueden incorporarse a una amplia gama de dispositivos, incluidos televisores digitales, sistemas de difusión directa digital, sistemas de difusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de sobremesa, ordenadores de tableta, cámaras digitales, dispositivos de grabación digitales, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, dispositivos de videoconferencia y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación Avanzada de Vídeo (AVC), la norma de Codificación de Vídeo de Alta Eficacia (HEVC), actualmente en desarrollo, y las extensiones de dichas normas, para transmitir, recibir y almacenar información de vídeo digital de forma más eficaz.

20

25

**[0004]** Las técnicas de compresión de vídeo incluyen la predicción espacial y/o la predicción temporal para reducir o eliminar la redundancia inherente en las secuencias de vídeo. Para la codificación de vídeo basada en bloques, una trama o un fragmento de vídeo puede dividirse en bloques. Una trama de vídeo, alternativamente, puede denominarse una imagen. Cada bloque se puede dividir adicionalmente. Los bloques en una imagen o fragmento intracodificado (I) se codifican usando la predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen o fragmento. Los bloques en una imagen o fragmento intercodificado (P o B) pueden usar la predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen o fragmento, o la predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. La predicción espacial o temporal da como resultado un bloque predictivo para un bloque a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original a codificar y el bloque predictivo.

30

35

**[0005]** Un bloque intercodificado se codifica de acuerdo a un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo a una modalidad de intracodificación y a los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio de píxeles a un dominio de transformación, dando como resultado coeficientes de transformación residuales, que a continuación se pueden cuantizar. Los coeficientes de transformación cuantizados, inicialmente dispuestos en una matriz bidimensional, pueden recorrerse en un orden particular con el fin de producir un vector unidimensional de coeficientes de transformación para la codificación por entropía. La codificación por entropía también se puede aplicar a otros varios elementos sintácticos utilizados en el proceso de codificación de vídeo. El documento de MARPE D ET AL: "Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard [Codificación aritmética binaria adaptativa basada en el contexto en la norma de compresión de vídeo H.264/AVC]", TRANSACCIONES DEL IEEE EN CIRCUITOS Y SISTEMAS PARA TECNOLOGÍA DE VÍDEO, vol. 13, no. 7, 1 de julio de 2003 (2003-07-01), páginas 620-636, XP002509017, ISSN: 1051-8215, DOI: 10.1109/TCSVT.2003.815173 divulga la codificación aritmética binaria adaptativa basada en el contexto (CABAC). Al combinar una técnica de codificación aritmética binaria adaptativa con el modelado del contexto, se logra un alto grado de adaptación y reducción de redundancia. También divulga un procedimiento de baja complejidad para la codificación aritmética binaria y la estimación de probabilidad que está bien adaptado para implementaciones eficaces de hardware y software.

40

45

50

55

## SUMARIO

60 **[0006]** La presente invención está definida en las reivindicaciones adjuntas, a las cuales se deberá hacer referencia a continuación.

65 **[0007]** Las técnicas de esta divulgación generalmente se refieren a la codificación por entropía de datos de vídeo. Por ejemplo, al realizar la codificación adaptativa por contexto, un codificador de vídeo puede codificar cada bit o "binario" de datos utilizando estimaciones de probabilidad, lo que puede indicar una probabilidad de que un binario tenga un valor binario determinado. Las estimaciones de probabilidad pueden incluirse dentro de un modelo de

probabilidad, también denominado "modelo de contexto". Un codificador de vídeo puede seleccionar un modelo de contexto determinando un contexto para el binario. El contexto para un binario puede incluir valores de binarios relacionados de elementos sintácticos previamente codificados. Después de codificar el binario, el codificador de vídeo puede actualizar el modelo de contexto basándose en un valor del binario para reflejar las estimaciones de probabilidad más actuales. A diferencia de la aplicación de una modalidad de codificación por contexto, un codificador de vídeo puede aplicar una modalidad de codificación por omisión. Por ejemplo, el codificador de vídeo puede usar una modalidad de omisión para evitar u omitir el proceso normal de codificación aritmética. En tales casos, un codificador de vídeo puede usar un modelo de probabilidad fijo (que no se actualiza durante la codificación) para codificar por omisión los binarios.

**[0008]** Las técnicas de esta divulgación se refieren a elementos sintácticos de codificación eficaz por contexto, asociados a datos de vídeo intercodificados. Por ejemplo, los aspectos de esta divulgación se refieren a la codificación eficaz de valores de índices de referencia, predictores de vectores de movimiento, valores de diferencia de vectores de movimiento y similares. En algunos casos, un codificador de vídeo puede realizar la codificación por contexto para algunos binarios de un elemento sintáctico y la codificación por omisión para otros binarios del elemento sintáctico. Por ejemplo, el codificador de vídeo puede codificar por contexto uno o más binarios de un valor de índice de referencia y codificar por omisión uno o más binarios del valor de índice de referencia.

**[0009]** En un ejemplo, los aspectos de esta divulgación se refieren a un procedimiento para codificar un elemento sintáctico de índice de referencia en un proceso de codificación de vídeo que incluye binarizar un valor de índice de referencia, codificar al menos un binario del valor de índice de referencia binarizado con una modalidad de codificación por contexto de un proceso de codificación aritmética binaria adaptable al contexto (CABAC), y codificar, cuando el valor del índice de referencia binarizado comprende más binarios que el al menos un binario codificado con la modalidad codificada por contexto, al menos otro binario del valor del índice de referencia binarizado con una modalidad de codificación por omisión del proceso de CABAC.

**[0010]** En otro ejemplo, los aspectos de esta divulgación se refieren a un aparato para codificar un elemento sintáctico de índice de referencia en un proceso de codificación de vídeo que incluye uno o más procesadores para binarizar un valor de índice de referencia, codificar al menos un binario del valor de índice de referencia binarizado con una modalidad de codificación por contexto de un proceso de codificación aritmética binaria adaptable al contexto (CABAC), y codificar, cuando el valor del índice de referencia binarizado comprende más binarios que el al menos un binario codificado con la modalidad codificada por contexto, al menos otro binario del valor del índice de referencia binarizado con una modalidad de codificación por omisión del proceso de CABAC.

**[0011]** En otro ejemplo, los aspectos de esta divulgación se refieren a un aparato para codificar un elemento sintáctico de índice de referencia en un proceso de codificación de vídeo que incluye medios para binarizar un valor de índice de referencia, medios para codificar al menos un binario del valor de índice de referencia binarizado con una modalidad de codificación por contexto de un proceso de codificación aritmética binaria adaptativa al contexto (CABAC), y medios para codificar, cuando el valor del índice de referencia binarizado comprende más binarios que el al menos un binario codificado con la modalidad codificada por contexto, al menos otro binario del valor del índice de referencia binarizado con una modalidad de codificación por omisión del proceso de CABAC.

**[0012]** En otro ejemplo, los aspectos de esta divulgación se refieren a un procedimiento para decodificar un elemento sintáctico de índice de referencia en un proceso de decodificación de vídeo que incluye la decodificación de al menos un binario de un valor de índice de referencia con una modalidad de codificación por contexto de un proceso de codificación aritmética binaria adaptada al contexto (CABAC), la decodificación, cuando el valor del índice de referencia comprende más binarios que el al menos un binario codificado con la modalidad codificada por contexto, de al menos otro binario del valor del índice de referencia con una modalidad de codificación por omisión del proceso de CABAC, y la binarización del valor del índice de referencia.

**[0013]** En otro ejemplo, los aspectos de esta divulgación se refieren a un aparato para decodificar un elemento sintáctico de índice de referencia en un proceso de decodificación de vídeo que incluye uno o más procesadores configurados para decodificar al menos un binario de un valor de índice de referencia con una modalidad de codificación por contexto de un proceso de codificación aritmética binaria adaptativa al contexto (CABAC), decodificar, cuando el valor del índice de referencia comprende más binarios que el al menos un binario codificado con la modalidad de codificación por contexto, al menos otro binario del valor del índice de referencia con una modalidad de omisión de codificación del proceso de CABAC, y binarizar el valor del índice de referencia.

**[0014]** En otro ejemplo, los aspectos de esta divulgación se refieren a un medio no transitorio legible por ordenador que almacena instrucciones en el mismo que, cuando se ejecutan, hacen que uno o más procesadores decodifiquen al menos un binario de un valor de índice de referencia con una modalidad de codificación por contexto de un proceso de codificación aritmética binaria adaptativa al contexto (CABAC), decodifiquen, cuando el valor del índice de referencia comprende más binarios que el al menos un binario codificado con la modalidad de codificación por contexto, al menos otro binario del valor del índice de referencia con una modalidad de codificación por omisión del proceso de CABAC, y binaricen el valor del índice de referencia.

**[0015]** Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la siguiente descripción. Otras características, objetos y ventajas resultarán evidentes a partir de la descripción y de los dibujos, y a partir de las reivindicaciones.

5

**BREVE DESCRIPCIÓN DE LOS DIBUJOS**

**[0016]**

10 La figura 1 es un diagrama de bloques que ilustra un sistema ejemplar de codificación y decodificación de vídeo.

La figura 2 es un diagrama de bloques que ilustra un codificador ejemplar de vídeo.

15 La figura 3 es un diagrama de bloques que ilustra un decodificador ejemplar de vídeo.

La figura 4 es un diagrama de bloques que ilustra un proceso ejemplar de codificación aritmética.

20 La figura 5A es un diagrama de bloques que ilustra una cadena ejemplar de datos de predicción.

La figura 5B es un diagrama de bloques que ilustra otra cadena ejemplar de datos de predicción.

La figura 6 es un diagrama de bloques que ilustra otra cadena ejemplar de datos de predicción.

25 La figura 7 es un diagrama de bloques que ilustra otra cadena ejemplar de datos de predicción.

La figura 8A es un diagrama de bloques que ilustra la codificación por contexto de un elemento sintáctico de dirección de interpredicción con tres valores posibles.

30 La figura 8B es un diagrama de bloques que ilustra la codificación por omisión del elemento sintáctico de dirección de interpredicción, de acuerdo a aspectos de esta divulgación.

La figura 9 es un diagrama de flujo que ilustra un ejemplo de codificación por entropía de un valor de índice de referencia, de acuerdo a aspectos de esta divulgación.

35

La figura 10 es un diagrama de flujo que ilustra un ejemplo de decodificación por entropía de un valor de índice de referencia, de acuerdo a aspectos de esta divulgación.

La figura 11 es un diagrama de flujo que ilustra un ejemplo de codificación por entropía de datos de predicción, de acuerdo a aspectos de esta divulgación.

40

La figura 12 es un diagrama de flujo que ilustra un ejemplo de decodificación por entropía de datos de predicción, de acuerdo a aspectos de esta divulgación.

45 **DESCRIPCIÓN DETALLADA**

**[0017]** Un dispositivo de codificación de vídeo puede comprimir datos de vídeo aprovechando la redundancia espacial y temporal. Por ejemplo, un codificador de vídeo puede aprovechar la redundancia espacial codificando un bloque con respecto a los bloques contiguos, codificados previamente. Del mismo modo, un codificador de vídeo puede aprovechar la redundancia temporal codificando un bloque en relación con datos de imágenes codificadas previamente. En particular, el codificador de vídeo puede predecir un bloque actual a partir de los datos de un vecino espacial (denominado intracodificación) o de los datos de otras una o más imágenes (denominado inter-codificación). A continuación, el codificador de vídeo calcula un valor residual para el bloque como una diferencia entre los valores reales de píxeles para el bloque y los valores de píxeles predichos para el bloque. En consecuencia, los datos residuales para un bloque incluyen valores de diferencia de píxel por píxel en el dominio de píxeles (o espacial).

50

55

**[0018]** Un codificador de vídeo puede realizar una estimación de movimiento y una compensación de movimiento al interpredecir un bloque de datos de vídeo. Por ejemplo, la estimación de movimiento se realiza en un codificador de vídeo e incluye el cálculo de uno o más vectores de movimiento. Un vector de movimiento puede indicar el desplazamiento de un bloque de datos de vídeo en una imagen actual, en relación con una muestra de referencia de una imagen de referencia. Una muestra de referencia puede ser un bloque del que se descubra que se corresponde estrechamente con el bloque que se está codificando, en términos de diferencia de píxeles, lo que puede determinarse mediante la suma de una diferencia absoluta (SAD), la suma de diferencias al cuadrado (SSD) u otras métricas de diferencia. La muestra de referencia puede producirse en cualquier lugar dentro de una imagen de referencia o de un fragmento de referencia, y no necesariamente en un límite de bloque de la imagen o

60

65

fragmento de referencia. En algunos ejemplos, la muestra de referencia puede producirse en una posición de píxel fraccionaria.

5 **[0019]** Los datos que definen el vector de movimiento pueden describir, por ejemplo, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, precisión de un cuarto de píxel o precisión de un octavo de píxel), una imagen de referencia a la que apunta el vector de movimiento y/o una lista de imágenes de referencia (por ejemplo, la Lista 0, la Lista 1 o la Lista C) para el vector de movimiento, que puede indicarse mediante una dirección de predicción. Un índice de referencia (ref\_idx) puede identificar la imagen en particular en la lista de imágenes de referencia a la que apunta el vector de movimiento. De esta manera, el elemento sintáctico ref\_idx sirve como un índice en una lista de imágenes de referencia, por ejemplo, L0, L1 o LC.

15 **[0020]** Tras la identificación de un bloque de referencia, se determina la diferencia entre el bloque de datos de vídeo original y el bloque de referencia. Esta diferencia puede denominarse datos residuales de predicción, e indica las diferencias de píxeles entre los valores de píxel en el bloque que vaya a codificarse y los valores de píxel en el bloque de referencia seleccionado para representar al bloque codificado. Para lograr una mejor compresión, los datos residuales de predicción pueden transformarse, por ejemplo, usando una transformación de coseno discreta (DCT), una transformación entera, una transformación de Karhunen-Loeve (KL) u otra transformación. Para una compresión adicional, los coeficientes de transformación pueden cuantizarse.

20 **[0021]** Un codificador de entropía luego codifica por entropía símbolos o elementos sintácticos asociados a un bloque de datos de vídeo y a los coeficientes de transformación cuantizados. Los ejemplos de esquemas de codificación por entropía incluyen la Codificación de Longitud Variable Adaptativa al Contexto (CAVLC), la Codificación Aritmética Binaria Adaptativa al Contexto (CABAC), la Codificación por Entropía de División en Intervalos de probabilidad (PIPE), o similares. Antes de la codificación por contexto, un codificador de vídeo puede convertir un valor absoluto de cada valor que se codifica en forma binarizada. De esta manera, cada valor no nulo que se codifica puede ser "binarizado", por ejemplo, utilizando una tabla de codificación unaria u otro esquema de codificación que convierta un valor en una palabra de código que tenga uno o más bits, o "binarios".

30 **[0022]** Con respecto a la CABAC, como ejemplo, el codificador de vídeo puede seleccionar un modelo de probabilidad (también denominado modelo de contexto) para codificar símbolos asociados a un bloque de datos de vídeo. Por ejemplo, en el codificador, un símbolo de destino puede codificarse utilizando el modelo de probabilidad. En el decodificador, se puede analizar sintácticamente un símbolo de destino utilizando el modelo de probabilidad. En algunos casos, se puede codificar binarios usando una combinación de codificación adaptativa al contexto y no adaptativa al contexto. Por ejemplo, un codificador de vídeo puede usar una modalidad de omisión para evitar u omitir el proceso normal de codificación aritmética para uno o más binarios, mientras usa la codificación adaptada al contexto para otros binarios. En dichos ejemplos, el codificador de vídeo puede usar un modelo de probabilidad fijo para codificar por omisión los binarios. Es decir, los binarios codificados por omisión no incluyen actualizaciones de contexto o probabilidad. En general, como se describe con mayor detalle con respecto a la figura 4 a continuación, la codificación de binarios por contexto puede denominarse codificación de binarios utilizando una modalidad de codificación por contexto. Del mismo modo, la codificación de binarios por omisión puede denominarse codificación de los binarios utilizando una modalidad de codificación por omisión.

45 **[0023]** Un modelo de contexto para codificar un binario de un elemento sintáctico puede basarse en valores de binarios relacionadas de elementos sintácticos vecinos codificados previamente. Como ejemplo, un modelo de contexto para codificar un binario de un elemento sintáctico actual puede basarse en valores de binarios relacionadas de elementos sintácticos vecinos codificados previamente, por ejemplo, en la parte superior y a la izquierda del elemento sintáctico actual. Las posiciones a partir de las cuales se obtiene el contexto pueden denominarse vecindario de soporte de contexto (también denominadas "soporte de contexto" o, simplemente, "soporte"). Por ejemplo, con respecto a la codificación de los binarios de un mapa de significación (por ejemplo, que indica las ubicaciones de los coeficientes de transformación no nulos en un bloque de datos de vídeo), se puede usar un soporte de cinco puntos para definir un modelo de contexto.

55 **[0024]** Por ejemplo, un modelo de contexto (Ctx) puede ser un índice o desplazamiento que se aplica para seleccionar uno entre una pluralidad de contextos diferentes, cada uno de los cuales puede corresponder a un modelo de probabilidad particular. En consecuencia, en todo caso se define habitualmente un modelo de probabilidad diferente para cada contexto. Después de codificar el binario, el modelo de probabilidad se actualiza adicionalmente basándose en un valor del binario para reflejar las estimaciones de probabilidad más recientes para el binario. Por ejemplo, un modelo de probabilidad puede mantenerse como un estado en una máquina de estados finitos. Cada estado particular puede corresponder a un valor de probabilidad específico. El siguiente estado, que corresponde a una actualización del modelo de probabilidad, puede depender del valor del binario actual (por ejemplo, el binario que se está codificando actualmente). Por consiguiente, la selección de un modelo de probabilidad puede estar influida por los valores de los binarios previamente codificados, porque los valores indican, al menos en parte, la probabilidad de que el binario tenga un valor dado. El proceso de codificación del contexto, descrito anteriormente, puede denominarse generalmente una modalidad de codificación adaptativa al contexto.

- 5 **[0025]** Además, el proceso de actualización de probabilidad, descrito anteriormente, puede introducir demoras en el proceso de codificación. Por ejemplo, supóngase que dos binarios utilizan el mismo modelo de contexto (por ejemplo, ctx (0)) con fines de codificación adaptativa al contexto. En este ejemplo, un primer binario puede usar ctx (0) para determinar un modelo de probabilidad para la codificación. El valor del primer binario influye en el modelo de probabilidad asociado a ctx (0). En consecuencia, se debe realizar una actualización de probabilidad antes de codificar el segundo binario con ctx (0). De esta forma, la actualización de probabilidad puede introducir un retraso en el ciclo de codificación.
- 10 **[0026]** Con respecto a la codificación de vídeo, como otro ejemplo, un codificador de vídeo puede codificar de manera adaptativa al contexto una secuencia de binarios (por ejemplo, bin (0), bin (1),... bin (n)) de un índice de referencia (ref\_idx). Como se señaló anteriormente, un índice de referencia (ref\_idx) puede identificar la imagen en particular en la lista de imágenes de referencia a la que apunta el vector de movimiento. Un solo índice de referencia (ref\_idx) puede incluir, por ejemplo, hasta 15 binarios. Supongamos, a los efectos de la explicación, que el codificador de vídeo obtiene tres contextos para codificar los binarios y aplica los contextos en función del número de binario que se está codificando (por ejemplo, indicado mediante los índices de contexto ctx (0), ctx (1) y ctx (2)). Es decir, en este ejemplo, el codificador de vídeo puede usar ctx (0) para codificar bin (0), ctx (1) para codificar bin (1) y ctx (2) para codificar los binarios restantes (por ejemplo, bin (2) hasta bin (n)).
- 20 **[0027]** En el ejemplo descrito anteriormente, el tercer contexto (ctx (2)) se comparte entre una serie de binarios (por ejemplo, hasta 13 binarios). El uso del mismo modelo de probabilidad para codificar desde bin (2) hasta bin(n) de esta manera puede crear un retraso entre ciclos de codificación sucesivos. Por ejemplo, como se ha señalado anteriormente, llamar repetidamente al mismo contexto y esperar a actualizar el modelo después de cada binario puede presentar un cuello de botella para el caudal del codificador.
- 25 **[0028]** Además, la correlación entre bin(2) y bin(n) puede no ser suficiente para garantizar el tiempo y los recursos informáticos asociados a la actualización del modelo de probabilidad. Es decir, un beneficio potencial de la codificación adaptativa al contexto es la capacidad de adaptar un modelo de probabilidad basándose en binarios previamente codificados (dado el mismo contexto). Sin embargo, si el valor de un primer binario tiene poca relación con, o relevancia en, el valor de un binario posterior, puede haber poca ganancia de eficacia asociada a la actualización de probabilidad. En consecuencia, los binarios que exhiben una baja correlación pueden no beneficiarse tanto de la codificación adaptativa al contexto como los binarios que tienen correlaciones relativamente mayores.
- 30 **[0029]** Los aspectos de esta divulgación se refieren a elementos sintácticos de codificación eficaz de contexto, asociados a datos de vídeo intercodificados. Por ejemplo, los aspectos de esta divulgación se refieren a la codificación eficaz de valores de índices de referencia, predictores de vectores de movimiento, valores de diferencia de vectores de movimiento y similares. En algunos ejemplos, un codificador de vídeo puede realizar la codificación por contexto para algunos binarios de un elemento sintáctico y la codificación por omisión para otros binarios del elemento sintáctico.
- 35 **[0030]** Con referencia específica al ejemplo de codificación de índice de referencia descrito anteriormente, según los aspectos de esta divulgación, el codificador de vídeo puede aplicar ctx (0) a bin (0), ctx (1) a bin (1), ctx (2) a bin (2), y puede omitir el código de los binarios restantes del valor del índice de referencia sin necesidad de contextos. En otras palabras, el codificador de vídeo puede usar ctx (2) como contexto para la codificación por CABAC de bin (2) de un valor de índice de referencia binarizado, pero puede codificar por omisión el código de cualquier binario que siga a bin (2).
- 40 **[0031]** Dado que los valores del índice de referencia pueden tener una longitud de 15 binarios o más, la limitación del número de binarios que están codificados por contexto de esta manera puede producir un ahorro de cálculo y/o de tiempo en comparación con la codificación por contexto de todos los binarios del índice de referencia. Además, como se ha señalado anteriormente, la correlación entre los bits de un valor de índice de referencia puede no ser alta (por ejemplo, el valor de bin (3) del valor de índice de referencia puede no proporcionar una indicación útil con respecto a la probabilidad de que bin (4) tenga un valor de un "1" o un "0"), lo que reduce el beneficio de la codificación por contexto. Por consiguiente, la cantidad de tiempo y recursos de cálculo ahorrados codificando por contexto menos binarios de un valor de índice de referencia puede compensar las ganancias de eficacia de codificación asociadas a la codificación por contexto de todos los binarios del valor de índice de referencia.
- 45 **[0032]** Otros aspectos de esta divulgación generalmente se refieren a la agrupación de binarios codificados por contexto y binarios no codificados por contexto durante la codificación. Por ejemplo, como se ha señalado anteriormente, algunos elementos sintácticos pueden codificarse utilizando una combinación de codificación por contexto y codificación por omisión. Es decir, algunos elementos sintácticos pueden tener uno o más binarios que están codificados por contexto y otros uno o más binarios que están codificados por omisión.
- 50 **[0033]** Supóngase, a modo de ejemplo, que dos elementos sintácticos tienen, cada uno, una parte codificada por contexto (que incluye uno o más binarios codificadas por contexto) y una parte codificada por omisión (incluidos
- 55
- 60
- 65

uno o más binarios codificados por omisión). En este ejemplo, el codificador de vídeo puede codificar la parte codificada por contexto del primer elemento sintáctico, seguida de la parte codificada por omisión del primer elemento sintáctico, seguida de la parte codificada por contexto del segundo elemento sintáctico, seguida de la parte codificada por omisión del segundo elemento sintáctico.

5 **[0034]** En el ejemplo descrito anteriormente, un codificador de vídeo puede conmutar, entre una modalidad de codificación por contexto y una modalidad de codificación por omisión, tres veces para codificar los dos elementos sintácticos. Por ejemplo, el codificador de vídeo conmuta entre la codificación por contexto y la codificación por omisión después de los binarios codificados por contexto del primer elemento sintáctico, después de los binarios codificados por omisión del primer elemento sintáctico, y después de los binarios codificados por contexto del segundo elemento sintáctico. La conmutación entre la codificación por contexto y la codificación por omisión de esta manera puede ser ineficaz en términos de cálculo. Por ejemplo, la conmutación entre la codificación por contexto y la codificación por omisión puede consumir uno o más ciclos de reloj. En consecuencia, la conmutación entre la codificación por contexto y la codificación por omisión para cada elemento puede introducir latencia, debido a las transiciones entre la codificación por contexto y la codificación por omisión.

10 **[0035]** Los aspectos de esta divulgación incluyen agrupar los binarios codificados por contexto y los binarios no codificados por contexto (por ejemplo, los binarios por omisión) durante la codificación. Por ejemplo, con respecto al ejemplo descrito anteriormente, de acuerdo a los aspectos de esta divulgación, un codificador de vídeo puede codificar los binarios codificados por contexto del primer elemento sintáctico, seguido de los binarios codificados por contexto del segundo elemento sintáctico, seguido de los binarios codificados por omisión del primer elemento sintáctico, seguido de los binarios codificados por omisión del segundo elemento sintáctico. Por consiguiente, el codificador de vídeo solo realiza transiciones entre una modalidad de codificación por contexto y una modalidad de codificación por omisión una sola vez, por ejemplo, entre los binarios codificados por contexto y los binarios no codificados por contexto.

20 **[0036]** Agrupar los binarios de esta manera puede reducir la frecuencia con la que un codificador de vídeo conmuta entre una modalidad de codificación por contexto y una modalidad de codificación por omisión. En consecuencia, los aspectos de esta divulgación pueden reducir la latencia cuando se codifican elementos sintácticos que incluyen una combinación de binarios codificados por contexto y binarios codificados por omisión. En algunos ejemplos, como se describe con respecto a las figuras 5 a 8 a continuación, los binarios asociados a los datos de predicción se pueden agrupar de acuerdo a las técnicas de esta divulgación. Por ejemplo, como se describe en el presente documento, los datos de predicción generalmente pueden incluir datos asociados a la interpredicción. Por ejemplo, los datos de predicción pueden incluir datos que indiquen valores de índice de referencia, vectores de movimiento, predictores de vectores de movimiento, valores de diferencia de vectores de movimiento y similares.

30 **[0037]** La figura 1 es un diagrama de bloques que ilustra un sistema ejemplar de codificación y decodificación de vídeo 10 que puede configurarse para codificar datos de predicción de acuerdo a ejemplos de esta divulgación. Como se muestra en la figura 1, el sistema 10 incluye un dispositivo de origen 12 que transmite vídeo codificado a un dispositivo de destino 14 mediante un canal de comunicación 16. Los datos de vídeo codificados también se pueden almacenar en un medio de almacenamiento 34 o en un servidor de ficheros 36, y se puede acceder a los mismos mediante el dispositivo de destino 14, según se desee. Cuando se almacene en un medio de almacenamiento o en un servidor de ficheros, el codificador de vídeo 20 puede proporcionar datos de vídeo codificados a otro dispositivo, tal como una interfaz de red, un grabador, o dispositivo utilitario de estampado, de disco compacto (CD), Blu-ray o discos de vídeo digital (DVD), u otros dispositivos, para almacenar los datos de vídeo codificados en el medio de almacenamiento. Asimismo, un dispositivo independiente del decodificador de vídeo 30, tal como una interfaz de red, un lector de CD o DVD, o similares, puede recuperar los datos de vídeo codificados desde un medio de almacenamiento y proporcionar los datos recuperados al decodificador de vídeo 30.

40 **[0038]** El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera entre una amplia variedad de dispositivos, que incluyen ordenadores de sobremesa, ordenadores plegables (es decir, portátiles), ordenadores de tableta, decodificadores, equipos manuales de telefonía tales como los denominados teléfonos inteligentes, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos o similares. En muchos casos, dichos dispositivos pueden estar equipados para la comunicación inalámbrica. Por lo tanto, el canal de comunicación 16 puede comprender un canal inalámbrico, un canal cableado o una combinación de canales inalámbricos y cableados, adecuados para la transmisión de datos de vídeo codificados. De manera similar, puede accederse al servidor de ficheros 36 mediante el dispositivo de destino 14, a través de cualquier conexión de datos estándar, incluida una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión de Wi-Fi), una conexión cableada (por ejemplo, DSL, módem de cable, etc.) o una combinación de ambas que sea adecuada para acceder a datos de vídeo codificados, almacenados en un servidor de ficheros.

50 **[0039]** Las técnicas para la codificación de datos de predicción, de acuerdo a los ejemplos de esta divulgación, pueden aplicarse a la codificación de vídeo, como soporte de cualquiera entre una diversidad de aplicaciones de

multimedios, tales como difusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo por flujo continuo, por ejemplo, mediante Internet, codificación de vídeo digital para su almacenamiento en un medio de almacenamiento de datos, decodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 se puede configurar para prestar soporte a la transmisión de vídeo unidireccional o bidireccional, para dar soporte a aplicaciones tales como la transmisión continua de vídeo, la reproducción de vídeo, la difusión de vídeo y/o la videotelefonía.

**[0040]** En el ejemplo de la figura 1, el dispositivo de origen 12 incluye un origen de vídeo 18, un codificador de vídeo 20, un modulador/demodulador 22 y un transmisor 24. En el dispositivo de origen 12, el origen de vídeo 18 puede incluir un origen tal como un dispositivo de captura de vídeo, tal como una videocámara, un archivo de vídeo que contenga vídeo previamente capturado, una interfaz de alimentación de vídeo para recibir vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos de ordenador para generar datos de gráficos de ordenador como el vídeo de origen, o una combinación de dichos orígenes. Como ejemplo, si el origen de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videoteléfonos. Sin embargo, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general y pueden aplicarse a aplicaciones inalámbricas y/o cableadas, o a aplicaciones en las que los datos de vídeo codificados se almacenen en un disco local.

**[0041]** El vídeo capturado, pre-capturado o generado por ordenador puede ser codificado por el codificador de vídeo 20. La información de vídeo codificada puede ser modulada por el módem 22 de acuerdo a una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitida al dispositivo de destino 14 mediante el transmisor 24. El módem 22 puede incluir diversos mezcladores, filtros, amplificadores u otros componentes diseñados para la modulación de señales. El transmisor 24 puede incluir circuitos diseñados para transmitir datos, incluyendo amplificadores, filtros y una o más antenas.

**[0042]** El vídeo capturado, precapturado, o generado por ordenador que es codificado mediante el codificador de vídeo 20 también puede almacenarse en un medio de almacenamiento 34 o en un servidor de ficheros 36 para un consumo posterior. Los medios de almacenamiento 34 pueden incluir discos Blu-ray, DVD, CD-ROM, memoria flash o cualquier otro medio de almacenamiento digital adecuado para almacenar vídeo codificado. Entonces se puede acceder al vídeo codificado almacenado en el medio de almacenamiento 34 mediante el dispositivo de destino 14, para su decodificación y reproducción.

**[0043]** El servidor de ficheros 36 puede ser cualquier tipo de servidor capaz de almacenar vídeo codificado y transmitir ese vídeo codificado al dispositivo de destino 14. Los servidores ejemplares de ficheros incluyen un servidor de la Red (por ejemplo, para una sede de la Red), un servidor del FTP, dispositivos de almacenamiento conectado a la red (NAS), una unidad de disco local o cualquier otro tipo de dispositivo capaz de almacenar datos de vídeo codificados y transmitirlos a un dispositivo de destino. La transmisión de datos de vídeo codificados desde el servidor de ficheros 36 puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas. Puede accederse al servidor de ficheros 36 mediante el dispositivo de destino 14, a través de cualquier conexión de datos estándar, incluso una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión cableada (por ejemplo, DSL, módem de cable, Ethernet, USB, etc.), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados, almacenados en un servidor de ficheros.

**[0044]** El dispositivo de destino 14, en el ejemplo de la figura 1, incluye un receptor 26, un módem 28, un decodificador de vídeo 30 y un dispositivo de visualización 32. El receptor 26 del dispositivo de destino 14 recibe información por el canal 16, y el módem 28 demodula la información para producir un flujo de bits demodulado para el decodificador de vídeo 30. La información comunicada por el canal 16 puede incluir una diversidad de información sintáctica generada por el codificador de vídeo 20 para su uso por el decodificador de vídeo 30 en la codificación de los datos de vídeo. Dicha sintaxis también puede incluirse con los datos de vídeo codificados, almacenados en los medios de almacenamiento 34 o en el servidor de ficheros 36. Cada uno entre el codificador de vídeo 20 y el decodificador de vídeo 30 puede formar parte de un respectivo codificador-decodificador (CÓDEC) que sea capaz de codificar o decodificar datos de vídeo.

**[0045]** El dispositivo de visualización 32 puede estar integrado con, o ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también estar configurado para interconectarse con un dispositivo de visualización externo. En otros ejemplos, el dispositivo de destino 14 puede ser un dispositivo de visualización. En general, el dispositivo de visualización 32 muestra los datos de vídeo decodificados a un usuario, y puede comprender cualquiera entre varios dispositivos de visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

**[0046]** En el ejemplo de la figura 1, el canal de comunicación 16 puede comprender cualquier medio de comunicación inalámbrico o cableado, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física, o cualquier combinación de medios inalámbricos y cableados. El canal de comunicación 16

puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global tal como Internet. El canal de comunicación 16 representa en general a cualquier medio de comunicación adecuado o un conjunto de diferentes medios de comunicación, para transmitir datos de vídeo desde el dispositivo de origen 12 al dispositivo de destino 14, incluida cualquier combinación adecuada de medios por cable o inalámbricos. El canal de comunicación 16 puede incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

**[0047]** El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo a una norma de compresión de vídeo, tal como la norma de Codificación de Vídeo de Alta Eficacia (HEVC), actualmente en fase de desarrollo, y pueden ajustarse al Modelo de Prueba de HEVC (HM). De forma alternativa, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo a otras normas patentadas o industriales, tales como la norma ITU-T H.264, de forma alternativa denominada MPEG-4, Parte 10, Codificación de Vídeo Avanzada (AVC), o ampliaciones de dichas normas. Sin embargo, las técnicas de esta divulgación no están limitadas a ninguna norma de codificación particular. Otros ejemplos incluyen MPEG-2 e ITU-T H.263.

**[0048]** Aunque no se muestra en la figura 1, en algunos aspectos, tanto el codificador de vídeo 20 como el decodificador de vídeo 30 pueden estar integrados con un codificador y decodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, u otro tipo de hardware y software, para ocuparse de la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos independientes. Si procede, en algunos ejemplos, las unidades MUX-DEMUX pueden ajustarse al protocolo de multiplexador ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

**[0049]** Tanto el codificador de vídeo 20 como el decodificador de vídeo 30 puede implementarse como cualquiera entre varios circuitos codificadores o decodificadores adecuados, según corresponda, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), formaciones de compuertas programables in situ (FPGA), circuitos de lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Por consiguiente, las diversas unidades dentro del codificador de vídeo 20 y el decodificador de vídeo 30 también pueden ser implementadas por cualquiera entre una diversidad de tales elementos estructurales o combinaciones de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado no transitorio legible por ordenador, y ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno entre el codificador de vídeo 20 y el decodificador de vídeo 30 se puede incluir en uno o más codificadores o decodificadores, cualquiera de los cuales se puede integrar como parte de un codificador/decodificador (CÓDEC) combinado en un dispositivo respectivo.

**[0050]** Esta divulgación puede referirse en general al codificador de vídeo 20 que "señaliza" cierta información a otro dispositivo, tal como el decodificador de vídeo 30. Sin embargo, debería entenderse que el codificador de vídeo 20 puede señalar información asociando ciertos elementos sintácticos a varias partes codificadas de datos de vídeo. Es decir, el codificador de vídeo 20 puede "señalizar" datos almacenando ciertos elementos sintácticos en cabeceras de varias partes codificadas de datos de vídeo. En algunos casos, dichos elementos sintácticos pueden ser codificados y almacenados (por ejemplo, almacenados en un dispositivo de almacenamiento 32) antes de ser recibidos y decodificados por el decodificador de vídeo 30. Por lo tanto, el término "señalización" puede referirse, en general, a la comunicación de datos sintácticos u otros, usados para decodificar datos de vídeo comprimidos, ya sea que dicha comunicación se produzca en tiempo real o casi real o durante un intervalo de tiempo, tal como podría ocurrir cuando se almacenan elementos sintácticos en un medio en el momento de la codificación, que entonces pueden ser recuperados por un dispositivo de decodificación en cualquier momento después de haber sido almacenados en este medio.

**[0051]** Como se ha indicado anteriormente, el equipo JCT-VC está trabajando en el desarrollo de la norma HEVC. La iniciativa de normalización de la HEVC se basa en un modelo en evolución de un dispositivo de codificación de vídeo denominado modelo de prueba de la HEVC (HM). El HM supone varias capacidades adicionales de los dispositivos de codificación de vídeo respecto a los dispositivos existentes, de acuerdo a, por ejemplo, la norma ITU-T H.264/AVC. Esta divulgación usa habitualmente el término «bloque de vídeo» para referirse a un nodo de codificación de una CU. En algunos casos específicos, esta divulgación también puede usar el término «bloque de vídeo» para referirse a un bloque arbolado, es decir, una LCU o una CU, que incluye un nodo de codificación y unas PU y TU.

**[0052]** Una secuencia de vídeo incluye habitualmente una serie de tramas o imágenes de vídeo. Un grupo de imágenes (GOP) comprende, en general, una serie de una o más de las imágenes de vídeo. Un GOP puede incluir datos sintácticos en una cabecera del GOP, una cabecera de una o más de las imágenes, o en otras ubicaciones, que describen una serie de imágenes incluidas en el GOP. Cada fragmento de una imagen puede incluir datos sintácticos de fragmento que describen una modalidad de codificación para el fragmento respectivo. El codificador de vídeo 20 actúa habitualmente sobre bloques de vídeo dentro de fragmentos de vídeo individuales con el fin de codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un nodo de codificación dentro de una CU. Los bloques de vídeo pueden tener tamaños fijos o variables y pueden diferir en tamaño de acuerdo a una norma

de codificación especificada.

**[0053]** Como ejemplo, y como se ha indicado anteriormente, el HM presta soporte a la predicción en varios tamaños de PU (también denominados tipos de PU). Si se supone que el tamaño de una CU particular es  $2N \times 2N$ , el HM presta soporte a la intrapredicción en tamaños de PU de  $2N \times 2N$  o  $N \times N$  y a la interpredicción en tamaños de PU simétricas de  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$  o  $N \times N$ . El HM también presta soporte a la división asimétrica para la interpredicción en tamaños de PU de  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$  y  $nR \times 2N$ . En la división asimétrica, una dirección de una CU no está dividida, mientras que la otra dirección está dividida entre un 25% y un 75%. La parte de la CU correspondiente a la división del 25% está indicada por una «n» seguida por una indicación de «arriba», «abajo», «izquierda» o «derecha». Así pues, por ejemplo, « $2N \times nU$ » se refiere a una CU de tamaño  $2N \times 2N$  que está dividida horizontalmente, con una PU de tamaño  $2N \times 0,5N$  encima y una PU de tamaño  $2N \times 1,5N$  debajo. También son posibles otros tipos de división.

**[0054]** En la presente divulgación, « $N \times N$ » y «N por N» pueden usarse indistintamente para hacer referencia a las dimensiones en píxeles de un bloque de vídeo en términos de dimensiones verticales y horizontales, por ejemplo,  $16 \times 16$  píxeles o 16 por 16 píxeles. En general, un bloque de  $16 \times 16$  tendrá 16 píxeles en una dirección vertical ( $y = 16$ ) y 16 píxeles en una dirección horizontal ( $x = 16$ ). Asimismo, un bloque de tamaño  $N \times N$  tiene, en general, N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo. Los píxeles de un bloque se pueden disponer en filas y columnas. Además, no es necesario que los bloques tengan necesariamente el mismo número de píxeles en la dirección horizontal que en la dirección vertical. Por ejemplo, los bloques pueden comprender  $N \times M$  píxeles, donde M no es necesariamente igual a N.

**[0055]** Tras la codificación intrapredictiva o interpredictiva, usando las PU de una CU, el codificador de vídeo 20 puede calcular datos residuales para las TU de la CU. Las PU pueden comprender datos de píxeles en el dominio espacial (también denominado dominio del píxel) y las TU pueden comprender coeficientes en el dominio de la transformación tras la aplicación de una transformación, por ejemplo, una transformación discreta de coseno (DCT), una transformación entera, una transformación de ondículas o una transformación conceptualmente similar, a los datos de vídeo residuales. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen no codificada y valores de predicción correspondientes a las PU. El codificador de vídeo 20 puede formar las TU incluyendo los datos residuales para la CU y, a continuación, transformar las TU para generar coeficientes de transformación para la CU.

**[0056]** En algunos ejemplos, como se ha indicado anteriormente, las TU pueden definirse de acuerdo a un RQT. Por ejemplo, un RQT puede representar la manera en que se aplican transformaciones (por ejemplo, DCT, transformación entera, transformación de ondículas u otras una o más transformaciones) a las muestras de luma residuales y las muestras de croma residuales asociadas a un bloque de datos de vídeo. Es decir, como se ha señalado anteriormente, las muestras residuales correspondientes a una CU pueden subdividirse en unidades más pequeñas utilizando un RQT. En general, el RQT es una representación recursiva de la partición de una CU en las TU.

**[0057]** Tras la aplicación de cualquier transformación a datos residuales para generar coeficientes de transformación, el codificador de vídeo 20 puede realizar la cuantización de los coeficientes de transformación. La cuantización se refiere, en general, a un proceso en el que unos coeficientes de transformación se cuantizan para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando compresión adicional. El proceso de cuantización puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes. Por ejemplo, un valor de  $n$  bits puede redondearse hacia abajo hasta un valor de  $m$  bits durante la cuantización, donde  $n$  es mayor que  $m$ .

**[0058]** En algunos ejemplos, el codificador de vídeo 20 puede utilizar un orden de exploración predefinido para explorar los coeficientes de transformación cuantizados, para producir un vector serializado que se pueda codificar por entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar una exploración adaptativa. Después de explorar los coeficientes de transformación cuantizados para formar un vector unidimensional, el codificador de vídeo 20 puede realizar la codificación por entropía del vector unidimensional, por ejemplo, de acuerdo a la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa al contexto (CABAC), la codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), la codificación por entropía de división en intervalos de probabilidad (PIPE) u otra metodología de codificación por entropía. El codificador de vídeo 20 también puede realizar la codificación por entropía de los elementos sintácticos asociados a los datos de vídeo codificados, para su uso por el decodificador de vídeo 30 en la decodificación de los datos de vídeo. Una versión actual de la HEVC está diseñada para usar CABAC para la codificación por entropía.

**[0059]** En algunos casos, el codificador de vídeo 20 puede codificar elementos sintácticos usando una combinación de codificación adaptativa al contexto y no adaptativa al contexto. Por ejemplo, el codificador de vídeo 20 puede codificar por contexto binarios seleccionando un modelo de probabilidad o "modelo de contexto" que opera sobre el contexto para codificar los binarios. Por el contrario, el codificador de vídeo 20 puede codificar por omisión los binarios evitando u omitiendo el proceso de codificación aritmética normal cuando se codifican los

binarios. En dichos ejemplos, el codificador de vídeo 20 puede usar un modelo de probabilidad fijo para codificar por omisión los binarios.

5 **[0060]** Como se ha señalado anteriormente, el proceso de actualización del modelo de probabilidad, asociado a la codificación por contexto, puede introducir un retraso en el proceso de codificación. Por ejemplo, el codificador de vídeo 20 puede codificar por contexto una secuencia de binarios (por ejemplo, bin (0), bin (1),... bin (n)) de un índice de referencia (ref\_idx). Un solo índice de referencia (ref\_idx) puede incluir, por ejemplo, hasta 15 binarios. Supóngase, con fines de explicación, que el codificador de vídeo 20 obtiene tres contextos para codificar los binarios y aplica los contextos en función del número de binario que se está codificando (por ejemplo, indicado mediante los índices de contexto ctx (0), ctx (1) y ctx (2)). Es decir, en este ejemplo, el codificador de vídeo 20 puede usar ctx (0) para codificar bin (0), ctx (1) para codificar bin (1), y ctx (2) para codificar los binarios restantes (por ejemplo, bin (2) hasta bin (n)).

15 **[0061]** En el ejemplo descrito anteriormente, el tercer contexto (ctx (2)) se comparte entre una serie de binarios (por ejemplo, hasta 13 binarios). El uso del mismo modelo de probabilidad para codificar desde bin (2) hasta bin(n) de esta manera puede crear un retraso entre ciclos de codificación sucesivos. Por ejemplo, llamar repetidamente al mismo contexto y esperar a actualizar el modelo después de cada binario puede presentar un cuello de botella para el caudal del codificador.

20 **[0062]** Además, la correlación entre bin(2) y bin(n) puede no ser suficiente para garantizar el tiempo y los recursos informáticos asociados a la actualización del modelo de probabilidad. Es decir, un beneficio potencial de la codificación adaptativa al contexto es la capacidad de adaptar un modelo de probabilidad basándose en binarios previamente codificados (dado el mismo contexto). Sin embargo, si el valor de un primer binario tiene poca relación con, o relevancia en, el valor de un binario posterior, puede haber poca ganancia de eficacia asociada a la actualización de probabilidad. En consecuencia, los binarios que exhiben una baja correlación pueden no beneficiarse tanto de la codificación adaptativa al contexto como los binarios que tienen correlaciones relativamente mayores.

30 **[0063]** De acuerdo a los aspectos de esta divulgación, el codificador de vídeo 20 puede codificar un elemento sintáctico de índice de referencia codificando al menos un binario de un valor de índice de referencia binarizado con un proceso de codificación aritmética binaria adaptativa al contexto (CABAC), y codificando al menos otro binario del valor del índice de referencia binarizado con una modalidad de codificación por omisión del proceso de codificación aritmética binaria adaptable al contexto (CABAC).

35 **[0064]** En un ejemplo con fines de ilustración, el codificador de vídeo 20 puede aplicar ctx (0) a bin (0), ctx (1) a bin (1), ctx (2) a bin (2) y puede codificar por omisión los binarios restantes del valor del índice de referencia sin contextos requeridos. En otras palabras, el codificador de vídeo puede usar ctx (2) como contexto para la codificación por CABAC de bin (2) de un valor de índice de referencia binarizado, pero puede codificar por omisión el código de cualquier binario que siga a bin (2).

40 **[0065]** Dado que los valores del índice de referencia pueden tener una longitud de 15 binarios o más, la limitación del número de binarios que están codificados por contexto de esta manera puede producir un ahorro de cálculo y/o de tiempo en comparación con la codificación por contexto de todos los binarios del índice de referencia. Además, como se ha señalado anteriormente, la correlación entre los bits de un valor de índice de referencia puede no ser alta (por ejemplo, el valor de bin (3) del valor de índice de referencia puede no proporcionar una indicación útil con respecto a la probabilidad de que bin (4) tenga un valor de un "1" o un "0"), lo que reduce el beneficio de la codificación por contexto. Por consiguiente, la cantidad de tiempo y recursos de cálculo ahorrados codificando por contexto menos binarios de un valor de índice de referencia puede compensar las ganancias de eficacia de codificación asociadas a la codificación por contexto de todos los binarios del valor de índice de referencia.

50 **[0066]** De acuerdo a otros aspectos de esta divulgación, el codificador de vídeo 20 puede agrupar binarios codificados por contexto y binarios no codificados por contexto durante la codificación. Por ejemplo, como se ha señalado anteriormente, algunos elementos sintácticos pueden codificarse utilizando una combinación de codificación por contexto y codificación por omisión. Es decir, algunos elementos sintácticos pueden tener uno o más binarios que están codificados por contexto y otros uno o más binarios que están codificados por omisión.

60 **[0067]** En algunos ejemplos, el codificador de vídeo 20 puede conmutar entre la codificación por contexto y la codificación por omisión para codificar una secuencia de elementos sintácticos. Sin embargo, la conmutación entre la codificación por contexto y la codificación por omisión puede consumir uno o más ciclos de reloj. En consecuencia, la conmutación entre la codificación por contexto y la codificación por omisión para cada elemento puede introducir latencia, debido a las transiciones entre la codificación por contexto y la codificación por omisión.

65 **[0068]** De acuerdo a los aspectos de esta divulgación, el codificador de vídeo 20 puede agrupar los binarios codificados por contexto y los binarios no codificados por contexto (por ejemplo, los binarios de omisión) durante la codificación. Por ejemplo, el codificador de vídeo 20 puede codificar por contexto los binarios asociados a más de un elemento sintáctico. El codificador de vídeo 20 puede entonces codificar por omisión los binarios asociados

a los más de uno elementos sintácticos. En otros ejemplos, el codificador de vídeo 20 puede realizar la codificación por omisión antes de la codificación por contexto. En cualquier caso, las técnicas permiten que el codificador de vídeo 20 minimice las transiciones entre la codificación por contexto y la codificación por omisión. En consecuencia, los aspectos de esta divulgación pueden reducir la latencia cuando se codifican elementos sintácticos que incluyen una combinación de binarios codificados por contexto y binarios codificados por omisión.

**[0069]** El decodificador de vídeo 30, al recibir los datos de vídeo codificados, puede llevar a cabo un pase de decodificación generalmente recíproco al pase de codificación descrito con respecto al codificador de vídeo 20. Por ejemplo, el decodificador de vídeo 30 puede recibir un flujo de bits codificado y decodificar el flujo de bits. De acuerdo a los aspectos de esta divulgación, por ejemplo, el decodificador de vídeo 30 puede decodificar un elemento sintáctico de índice de referencia, codificando al menos un binario de un valor de índice de referencia binarizado, con un proceso de codificación aritmética binaria adaptativa al contexto (CABAC), y codificando al menos otro binario del valor del índice de referencia binarizado, con una modalidad de codificación por omisión del proceso de codificación aritmética binaria adaptable al contexto (CABAC).

**[0070]** De acuerdo a otros aspectos de esta divulgación, el decodificador de vídeo 30 puede decodificar un flujo de bits que tenga agrupados binarios codificados por contexto y binario no codificados por contexto (por ejemplo, binarios de omisión). Por ejemplo, el decodificador de vídeo 30 puede decodificar los binarios codificados por contexto asociados a más de un elemento sintáctico. El decodificador de vídeo 30 puede entonces decodificar los binarios codificados por omisión, asociados a más de un elemento sintáctico. En otros ejemplos, el decodificador de vídeo 30 puede realizar la codificación por omisión antes de la codificación por contexto (según la disposición de los binarios en el flujo de bits que se está decodificando). En cualquier caso, las técnicas permiten que el decodificador de vídeo 30 minimice las transiciones entre la codificación por contexto y la codificación por omisión. En consecuencia, los aspectos de esta divulgación pueden reducir la latencia cuando se codifican elementos sintácticos que incluyen una combinación de binarios codificados por contexto y binarios codificados por omisión.

**[0071]** La figura 2 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo 20 que puede utilizar técnicas para codificar datos de predicción, de acuerdo a ejemplos de esta divulgación. Si bien los aspectos del codificador de vídeo 20 se describen en el contexto de la codificación HEVC con fines ilustrativos, las técnicas de esta divulgación no están limitadas a ninguna norma de codificación en particular, ni ningún procedimiento que pueda requerir la codificación de datos de predicción.

**[0072]** El codificador de vídeo 20 puede realizar intracodificación e intercodificación de las CU dentro de tramas de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en los datos de vídeo dentro de una imagen dada. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal entre una imagen actual e imágenes previamente codificadas de una secuencia de vídeo. La intramodalidad (modalidad I) puede referirse a cualquiera entre varias modalidades de compresión de vídeo de base espacial. Las intermodalidades tales como la predicción unidireccional (modalidad P) o la predicción bidireccional (modalidad B) pueden referirse a cualquiera entre varias modalidades de compresión de vídeo de base temporal.

**[0073]** Como se muestra en la figura 2, el codificador de vídeo 20 recibe un bloque de vídeo actual dentro de una imagen a codificar. En el ejemplo de la figura 2, el codificador de vídeo 20 incluye una unidad de compensación de movimiento 44, una unidad de estimación de movimiento 42, una unidad de intrapredicción 46, una memoria de imágenes de referencia 64, un sumador 50, una unidad de procesamiento de transformación 52, una unidad de cuantización 54 y una unidad de codificación por entropía 56. La unidad de procesamiento de transformación 52 ilustrada en la figura 2 es la unidad que aplica la transformación real, o combinaciones de transformaciones, a un bloque de datos residuales, y no ha de confundirse con un bloque de coeficientes de transformación, que también se puede denominar unidad de transformación (TU) de una CU. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye además una unidad de cuantización inversa 58, una unidad de procesamiento de transformación inversa 60 y un sumador 62. También se puede incluir un filtro de eliminación de bloques (no mostrado en la figura 2) para filtrar fronteras de bloques, para eliminar distorsiones de efecto pixelado del vídeo reconstruido. Si se desea, el filtro de eliminación de bloques filtrará habitualmente la salida del sumador 62.

**[0074]** Durante el proceso de codificación, el codificador de vídeo 20 recibe una imagen o un fragmento a codificar. La imagen o el fragmento pueden dividirse en múltiples bloques de vídeo, por ejemplo, unidades máximas de codificación (LCU). La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 llevan a cabo la codificación interpredictiva del bloque de vídeo recibido con respecto a uno o más bloques en una o más imágenes de referencia, para proporcionar compresión temporal. La unidad de intrapredicción 46 puede llevar a cabo una codificación intrapredictiva del bloque de vídeo recibido con respecto a uno o más bloques contiguos en la misma imagen o fragmento que el bloque a codificar, para proporcionar compresión espacial.

**[0075]** La unidad de selección de modalidad 40 puede seleccionar una de las modalidades de codificación (intra o inter), por ejemplo, en función de los resultados de error (es decir, de distorsión) para cada modalidad, y proporcionar el bloque intracodificado o intercodificado resultante (por ejemplo, una unidad de predicción (PU)) al sumador 50 para generar datos de bloque residuales, y al sumador 62 para reconstruir el bloque codificado para

su uso en una imagen de referencia. El sumador 62 combina el bloque predicho con los datos inversamente cuantizados, inversamente transformados, provenientes de la unidad de procesamiento de transformación inversa 60 para el bloque, para reconstruir el bloque codificado, como se describe en mayor detalle a continuación. Algunas imágenes pueden designarse como tramas I, donde todos los bloques en una trama I se codifican en una modalidad de intrapredicción. En algunos casos, la unidad de intrapredicción 46 puede realizar la codificación por intrapredicción de un bloque en una imagen predicha hacia adelante (trama P) o una imagen bipredicha (trama B), por ejemplo, cuando la búsqueda de movimiento realizada por la unidad de estimación de movimiento 42 no dé como resultado una predicción suficiente del bloque.

**[0076]** La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar sumamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento (o búsqueda de movimiento) es el proceso de generación de vectores de movimiento, que estiman el movimiento para los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una unidad de predicción en una trama actual con respecto a una muestra de referencia de una imagen de referencia. La unidad de estimación de movimiento 42 calcula un vector de movimiento para una unidad de predicción de una imagen intercodificada, mediante la comparación de la unidad de predicción con las muestras de referencia de una imagen de referencia almacenada en la memoria de imágenes de referencia 64.

**[0077]** Un bloque predictivo (también mencionado como una muestra de referencia) es un bloque que se revela como estrechamente coincidente con el bloque a codificar, en lo que respecta a la diferencia de píxeles, lo cual puede determinarse mediante la suma de diferencias absolutas (SAD), suma de las diferencias al cuadrado (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones fraccionarias de píxel de imágenes de referencia almacenadas en la memoria de imágenes de referencia 64, que también se puede denominar memoria intermedia de imágenes de referencia. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones fraccionarias de píxel de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento relativa a las posiciones de píxel completo y a las posiciones de píxel fraccionario, y emitir un vector de movimiento con una precisión de píxel fraccionario.

**[0078]** La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de vídeo en un fragmento intercodificado, comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. En general, los datos para un vector de movimiento pueden incluir una lista de imágenes de referencia, un índice para la lista de imágenes de referencia (ref\_idx), un componente horizontal y un componente vertical. La imagen de referencia se puede seleccionar a partir de una primera lista de imágenes de referencia (lista 0), una segunda lista de imágenes de referencia (lista 1) o una lista de imágenes de referencia combinada (lista C), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la memoria de imágenes de referencia 64.

**[0079]** La unidad de estimación de movimiento 42 puede generar y enviar un vector de movimiento que identifica el bloque predictivo de la imagen de referencia a la unidad de codificación por entropía 56 y a la unidad de compensación de movimiento 44. Es decir, la unidad de estimación de movimiento 42 puede generar y enviar datos de vectores de movimiento que identifican la lista de imágenes de referencia que contiene el bloque predictivo, un índice para la lista de imágenes de referencia que identifica la imagen del bloque predictivo y un componente horizontal y vertical para localizar el bloque predictivo dentro de la imagen identificada.

**[0080]** En algunos ejemplos, en lugar de enviar el vector de movimiento real para una PU actual, la unidad de estimación de movimiento 42 puede predecir el vector de movimiento para reducir aún más la cantidad de datos necesarios para comunicar el vector de movimiento. Por ejemplo, en lugar de codificar y comunicar el propio vector de movimiento, la unidad de estimación de movimiento 42 puede generar una diferencia de vector de movimiento (MVD) en relación con un vector de movimiento conocido (o que puede conocerse). La MVD puede incluir un componente horizontal y un componente vertical correspondiente al componente horizontal y el componente vertical del vector de movimiento conocido. El vector de movimiento conocido, que se puede usar con la MVD para definir el vector de movimiento actual, puede definirse mediante el llamado predictor de vector de movimiento (MVP). En general, para ser un MVP válido, el vector de movimiento que se usa para la predicción debe apuntar a la misma imagen de referencia que el vector de movimiento que se está codificando actualmente.

**[0081]** Cuando están disponibles múltiples candidatos de predictor de vector de movimiento (a partir de múltiples bloques candidatos), la unidad de estimación de movimiento 42 puede determinar un predictor de vector de movimiento para un bloque actual, de acuerdo a criterios de selección predeterminados. Por ejemplo, la unidad de estimación de movimiento 42 puede seleccionar el predictor más preciso del conjunto candidato basándose en el análisis de la tasa de codificación y distorsión (por ejemplo, utilizando un análisis de costes de velocidad-distorsión u otro análisis de eficacia de codificación). En otros ejemplos, la unidad de estimación de movimiento 42 puede generar un promedio de los candidatos predictores de vectores de movimiento. También son posibles otros procedimientos para seleccionar un predictor de vector de movimiento.

**[0082]** Al seleccionar un predictor de vector de movimiento, la unidad de estimación de movimiento 42 puede

determinar un índice de predictor de vector de movimiento (*mvp\_flag*), que puede usarse para informar a un decodificador de vídeo (por ejemplo, tal como el decodificador de vídeo 30) dónde ubicar el MVP en una lista de imágenes de referencia que contiene bloques candidatos a MVP. La unidad de estimación de movimiento 42 también puede determinar la MVD (componente horizontal y componente vertical) entre el bloque actual y el MVP seleccionado. El índice de MVP y la MVD se pueden usar para reconstruir el vector de movimiento.

**[0083]** En algunos ejemplos, la unidad de estimación de movimiento 42 puede, en cambio, implementar una llamada "modalidad de combinación", en la que la unidad de estimación de movimiento 42 puede "combinar" información de movimiento (tal como vectores de movimiento, índices de imágenes de referencia, direcciones de predicción u otra información) de un bloque de vídeo predictivo con un bloque de vídeo actual. Por consiguiente, con respecto a la modalidad de combinación, un bloque de vídeo actual hereda la información de movimiento desde otro bloque de vídeo conocido (o que puede conocerse). La unidad de estimación de movimiento 42 puede construir una lista de candidatos de la modalidad de combinación que incluya varios bloques vecinos en direcciones espaciales y/o temporales, como candidatos para la modalidad de combinación. La unidad de estimación de movimiento 42 puede determinar un valor de índice (por ejemplo, *merge\_idx*), que se puede usar para informar a un decodificador de vídeo (por ejemplo, tal como el decodificador de vídeo 30) dónde localizar el bloque de vídeo en combinación en una lista de imágenes de referencia que contiene bloques candidatos a combinación.

**[0084]** La unidad de intrapredicción 46 puede intrapredicir el bloque recibido, como alternativa a la interpredicción llevada a cabo por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44. La unidad de intrapredicción 46 puede predecir el bloque recibido respecto a bloques vecinos, previamente codificados, por ejemplo, bloques arriba, arriba y a la derecha, arriba y a la izquierda, o a la izquierda del bloque actual, suponiendo un orden de codificación de izquierda a derecha y de arriba a abajo para los bloques. La unidad de intrapredicción 46 se puede configurar con varias modalidades diferentes de intrapredicción. Por ejemplo, la unidad de intrapredicción 46 puede estar configurada con un cierto número de modalidades de predicción direccional, por ejemplo, treinta y cuatro modalidades de predicción direccional, basándose en el tamaño de la CU codificada.

**[0085]** La unidad de intrapredicción 46 puede seleccionar una modalidad de intrapredicción, por ejemplo, calculando valores de error para diversas modalidades de intrapredicción y seleccionando una modalidad que produzca el valor de error más bajo. Las modalidades de predicción direccional pueden incluir funciones para combinar valores de los píxeles espacialmente vecinos y aplicar los valores combinados a una o más posiciones de píxeles en una PU. Una vez que se han calculado los valores para todas las posiciones de píxel en la PU, la unidad de intra-predicción 46 puede calcular un valor de error para la modalidad de predicción basándose en diferencias de píxeles entre la PU y el bloque recibido para codificar. La unidad de intrapredicción 46 puede seguir probando modalidades de intrapredicción hasta que se descubra una modalidad de intrapredicción que produzca un valor de error aceptable. La unidad de intrapredicción 46 puede enviar luego la PU al sumador 50.

**[0086]** El codificador de vídeo 20 forma un bloque residual restando los datos de predicción calculados por la unidad de compensación de movimiento 44, o la unidad de intrapredicción 46, del bloque de vídeo original que se está codificando. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. El bloque residual puede corresponder a una matriz bidimensional de valores de diferencia de píxeles, donde el número de valores en el bloque residual es el mismo que el número de píxeles en la PU correspondiente al bloque residual. Los valores en el bloque residual pueden corresponder a las diferencias, es decir, al error, entre los valores de los píxeles ubicados conjuntamente en la PU y en el bloque original que a codificar. Las diferencias pueden ser diferencias de crominancia o de luminancia, según el tipo de bloque que se codifique.

**[0087]** La unidad de procesamiento de transformación 52 puede formar una o más unidades de transformación (TU) a partir del bloque residual. La unidad de procesamiento de transformación 52 selecciona una transformación entre una pluralidad de transformaciones. La transformación puede seleccionarse basándose en una o más características de codificación, tales como el tamaño de bloque, la modalidad de codificación o similares. A continuación, la unidad de procesamiento de transformación 52 aplica la transformación seleccionada a la TU, produciendo un bloque de vídeo que comprende una matriz bidimensional de coeficientes de transformación.

**[0088]** La unidad de procesamiento de transformación 52 puede enviar los coeficientes de transformación resultantes a la unidad de cuantización 54. La unidad de cuantización 54 puede cuantizar luego los coeficientes de transformación. La unidad de codificación por entropía 56 puede realizar luego un recorrido de los coeficientes de transformación cuantizados en la matriz, de acuerdo a una modalidad de recorrido. Esta divulgación describe la unidad de codificación por entropía 56 como realizadora del recorrido. Sin embargo, debería entenderse que, en otros ejemplos, otras unidades de procesamiento, tales como la unidad de cuantización 54, podrían realizar el recorrido.

**[0089]** Una vez que los coeficientes de transformación son recorridos para ingresar a la matriz unidimensional, la unidad de codificación por entropía 56 puede aplicar una codificación por entropía, tal como la CAVLC, la CABAC, la codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), la entropía por división en intervalos de probabilidad (PIPE), u otra metodología de codificación por entropía, a los coeficientes.

**[0090]** Para realizar la CABAC, la unidad de codificación por entropía 56 puede seleccionar un modelo de contexto para aplicar a un cierto contexto, para codificar los símbolos a transmitir. El contexto puede referirse, por ejemplo, a si los valores vecinos son distintos de cero o no. La unidad de codificación por entropía 56 también puede codificar por entropía elementos sintácticos, tales como la señal representativa de la transformación seleccionada.

**[0091]** La unidad de codificación por entropía 56 puede codificar por entropía los datos de predicción. Cuando los datos de vídeo se interpredicen, por ejemplo, los datos de predicción pueden incluir datos que indiquen valores de índice de referencia, vectores de movimiento, predictores de vectores de movimiento, valores de diferencia de vectores de movimiento y similares. Es decir, como se ha señalado anteriormente, la estimación de movimiento (por la unidad de estimación de movimiento 42) determina uno o más índices para las imágenes de referencia (ref\_idx) y una dirección de predicción (dir pred: hacia adelante, hacia atrás o bidireccional). La unidad de codificación por entropía 56 puede codificar por entropía elementos sintácticos que representan vectores de movimiento (por ejemplo, un componente horizontal y componente vertical de los vectores de movimiento), índices de imágenes de referencia y dirección de predicción. La unidad de codificación por entropía 56 puede incluir los elementos sintácticos codificados en un flujo de bits de vídeo codificado, que luego puede ser decodificado por un decodificador de vídeo (tal como el decodificador de vídeo 30, descrito a continuación) para su uso en un proceso de decodificación de vídeo. Es decir, estos elementos sintácticos pueden proporcionarse para una PU intercodificada, para permitir que el decodificador de vídeo 30 decodifique y reproduzca datos de vídeo definidos por una PU.

**[0092]** En algunos ejemplos, como se describe con mayor detalle con respecto a la figura 4 a continuación, la unidad de codificación por entropía 56 (u otra unidad de codificación del codificador de vídeo 20) puede binarizar los elementos sintácticos antes de la codificación por entropía de los elementos sintácticos. Por ejemplo, la unidad de codificación por entropía 56 puede convertir un valor absoluto de cada elemento sintáctico que se codifica en forma binaria. La unidad de codificación por entropía 56 puede usar un proceso de codificación unario, unario truncado u otro, para binarizar los elementos sintácticos. Con respecto a los valores de índices de referencia, por ejemplo, si un número máximo de imágenes de referencia en una lista de imágenes de referencia es cuatro, es decir, el índice de referencia (ref\_idx) tiene un valor que va de 0 a 3, entonces se puede aplicar la siguiente binarización en la Tabla 1:

**TABLA 1**

Índice de referencia	Binarización
0	0
1	10
2	110
3	111

Como se muestra en la Tabla 1, el valor binarizado varía de uno a tres bits, según el valor del índice de referencia.

**[0093]** En algunos ejemplos, la unidad de codificación por entropía 56 puede codificar por entropía los valores de índices de referencia utilizando tres contextos diferentes (por ejemplo, ctx0, ctx1 y ctx2). Por ejemplo, la unidad de codificación por entropía 56 puede codificar por entropía un primer binario (bin0) y un segundo binario (bin1) utilizando ctx0 y ctx1, respectivamente, y un tercer binario (bin2) y otros binarios están codificados con el contexto ctx2. En este ejemplo, ctx2 se comparte entre todos los binarios a partir de e incluyendo bin2, es decir, bin2 y binarios después de bin2, por ejemplo, bin3, bin4, etc. En algunos ejemplos, se pueden proporcionar binarios adicionales más allá de bin2, por ejemplo, si el número máximo de imágenes de referencia es mayor que cuatro.

**[0094]** Como se ha señalado anteriormente, compartir el contexto ctx2 entre binarios puede ser ineficaz, debido a las actualizaciones de probabilidad asociadas a la codificación por contexto. De acuerdo a los aspectos de esta descripción, la unidad de codificación por entropía 56 puede codificar por CABAC un valor de índice de referencia dedicando ctx2 a codificar bin2 y codificando todos los binarios después de bin2 usando una modalidad de codificación por omisión. De nuevo, la codificación por omisión generalmente incluye codificar binarios utilizando una probabilidad fija (no se requieren contextos). Por ejemplo, la unidad de codificación por entropía 56 puede codificar por omisión binarios, después del bin2, de un valor de índice de referencia utilizando la codificación de Golomb, la codificación de Golomb exponencial, la codificación de Golomb-Rice u otros procesos de codificación que evitan un motor de codificación CABAC.

**[0095]** En otro ejemplo, la unidad de codificación por entropía 56 puede codificar por contexto menos binarios de un valor de índice de referencia, mediante la eliminación de ctx2. Es decir, según los aspectos de esta divulgación, la unidad de codificación por entropía 56 puede codificar el bin2 y todas los binarios posteriores utilizando una modalidad de omisión de CABAC. En este ejemplo, la unidad de codificación por entropía 56 puede codificar por

CABAC el bin0 usando el contexto ctx0 y el bin1 utilizando el contexto ctx1, y puede codificar por omisión el bin2 y otros binarios que siguen a bin2 utilizando una modalidad de omisión de CABAC. Eliminar un contexto de esta manera puede reducir la complejidad general asociada a la codificación de valores de índices de referencia.

5 **[0096]** En otro ejemplo más, la unidad de codificación por entropía 56 puede codificar menos binarios de un valor de índice de referencia eliminando tanto ctx1 como ctx2. Es decir, según los aspectos de esta divulgación, la unidad de codificación por entropía 56 puede codificar el bin1 y todos los binarios posteriores utilizando una modalidad de omisión de CABAC, reduciendo por ello aún más la complejidad asociada a la codificación de valores de índices de referencia. En este ejemplo, la unidad de codificación por entropía 56 puede codificar por CABAC el bin0 usando el contexto ctx0, y puede codificar por omisión el bin1, el bin2 y otros binarios que siguen a bin2 utilizando una modalidad de omisión de CABAC.

15 **[0097]** Otros aspectos de esta divulgación se refieren generalmente a la manera en que la unidad de codificación por entropía 56 binariza los valores de índices de referencia. Por ejemplo, como se ha indicado anteriormente, la unidad de codificación por entropía 56 puede binarizar los valores de índices de referencia utilizando un proceso de codificación unario, unario truncado u otro. En otro ejemplo, la unidad de codificación por entropía 56 puede usar un proceso de codificación de Golomb exponencial para binarizar el valor del índice de referencia.

20 **[0098]** En algunos ejemplos, según los aspectos de esta divulgación, la unidad de codificación por entropía 56 puede implementar una combinación de procesos de binarización. Por ejemplo, como se describe con mayor detalle con respecto a la figura 4 a continuación, la unidad de codificación por entropía 56 puede combinar un proceso de codificación unario (o truncado unario) con un proceso de codificación de Golomb exponencial para binarizar los valores de índices de referencia. En un ejemplo con fines de ilustración, la unidad de codificación por entropía 56 puede combinar un código unario truncado de longitud (4) con un código de Golomb exponencial (por ejemplo, un código de Golomb exponencial de orden 0). En tal ejemplo, la unidad de codificación por entropía 56 puede binarizar un primer número de binarios (por ejemplo, los dos, tres, cuatro o similares) de un valor de índice de referencia utilizando un código unario, y puede binarizar los binarios restantes del índice de referencia utilizando un código de Golomb exponencial.

30 **[0099]** En cualquier caso, la unidad de codificación por entropía 56 puede implementar las técnicas para la codificación por contexto de uno o más binarios de un valor de índice de referencia y codificar por omisión uno o más binarios del valor de índice de referencia con cualquier esquema de binarización. Por ejemplo, como se ha indicado anteriormente, la unidad de codificación por entropía 56 puede codificar por contexto (por ejemplo, codificar por CABAC) un primer número de binarios de un elemento sintáctico binarizado y codificar los binarios restantes. En el ejemplo descrito anteriormente, en el que un código unario truncado de longitud (4) se combina con un código Golomb exponencial de orden 0, la unidad de codificación por entropía 56 puede codificar por contexto los dos primeros binarios (o cualquier otro número de binarios) del código unario truncado y luego el codificar por omisión la segunda parte del código unario y el código Golomb exponencial completo. En otros ejemplos, la unidad de codificación por entropía 56 puede usar otros esquemas de binarización. Por ejemplo, la unidad de codificación por entropía 56 puede usar un código binario de longitud fija en lugar del código de Golomb exponencial descrito en los ejemplos anteriores.

45 **[0100]** En algunos ejemplos, la unidad de codificación por entropía 56 puede trunca, o eliminar los binarios de un valor de índice de referencia binarizado antes de codificar el valor. Adicional o alternativamente, la unidad de codificación por entropía 56 puede agrupar los binarios codificados utilizando el contexto y los binarios codificados utilizando una modalidad de omisión. Por ejemplo, la unidad de codificación por entropía 56 puede codificar índices de referencia de una imagen B mediante codificación por contexto de uno o más binarios de un primer valor de índice de referencia, codificar por contexto uno o más binarios de un segundo valor de índice, codificar por omisión uno o más binarios del primer valor de índice de referencia, y codificar por omisión otros uno o más binarios del segundo valor de índice de referencia (en el orden presentado anteriormente). Por consiguiente, la unidad de codificación por entropía 56 solo realiza transiciones entre una modalidad de codificación por contexto y una modalidad de codificación por omisión una sola vez, por ejemplo, entre los binarios codificados por contexto y los binarios codificados sin contexto.

55 **[0101]** Tras la codificación por entropía mediante la unidad de codificación por entropía 56, el vídeo codificado resultante puede transmitirse a otro dispositivo, tal como el decodificador de vídeo 30, o archivar para su posterior transmisión o recuperación. La unidad de cuantización inversa 58 y la unidad de procesamiento de transformación inversa 60 aplican la cuantización inversa y la transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio del píxel, por ejemplo, para su posterior uso como bloque de referencia.

60 **[0102]** La unidad de compensación de movimiento 44 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de una de las imágenes de la memoria de imágenes de referencia 64. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores fraccionarios de píxel, para su uso en la estimación de movimiento.

65 **[0103]** El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado por movimiento,

producido por la unidad de compensación de movimiento 44, para producir un bloque de vídeo reconstruido para su almacenamiento en la memoria de imágenes de referencia 64. El bloque de vídeo reconstruido puede ser usado por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 como un bloque de referencia para intercodificar un bloque en una imagen posterior.

**[0104]** La figura 3 es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo 30, que decodifica una secuencia de vídeo codificado. En el ejemplo de la figura 3, el decodificador de vídeo 30 incluye una unidad de decodificación por entropía 70, una unidad de compensación de movimiento 72, una unidad de intrapredicción 74, una unidad de cuantización inversa 76, una unidad de transformación inversa 78, una memoria de imágenes de referencia 82 y un sumador 80.

**[0105]** A modo de referencia, el decodificador de vídeo 30 puede recibir datos de vídeo comprimidos que han sido comprimidos para su transmisión a través de una red a las llamadas "unidades de capa de abstracción de red" o unidades de NAL. Cada unidad de NAL incluye una cabecera que identifica un tipo de datos almacenados en la unidad de NAL. Hay dos tipos de datos que normalmente se almacenan en unidades de NAL. El primer tipo de datos almacenados en una unidad de NAL son los datos de la capa de codificación de vídeo (VCL), que incluyen los datos de vídeo comprimidos. El segundo tipo de datos almacenados en una unidad de NAL se menciona como datos no de VCL, que incluyen información adicional, tal como los conjuntos de parámetros que definen los datos de cabecera, comunes a un gran número de unidades de NAL e información de mejora complementaria (SEI).

**[0106]** Por ejemplo, los conjuntos de parámetros pueden contener la información de cabecera a nivel de secuencia (por ejemplo, en conjuntos de parámetros en secuencia (SPS)) y la información de cabecera a nivel de imagen, que cambia con poca frecuencia (en conjuntos de parámetros de imagen (PPS)). La información que cambia con poca frecuencia, contenida en los conjuntos de parámetros, no necesita repetirse para cada secuencia o imagen, mejorando por ello la eficacia de la codificación. Además, el uso de conjuntos de parámetros permite la transmisión fuera de banda de la información de cabecera, evitando por ello la necesidad de transmisiones redundantes para la recuperación frente a errores.

**[0107]** Durante el proceso de decodificación, el decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado, que representa bloques de vídeo de un fragmento de vídeo codificado, y elementos sintácticos asociados. En general, la unidad de decodificación por entropía 70 decodifica el flujo de bits para generar coeficientes cuantizados, vectores de movimiento y otros elementos sintácticos. El decodificador de vídeo 30 puede recibir los elementos sintácticos en el nivel de fragmento de vídeo y/o el nivel de bloque de vídeo.

**[0108]** Por ejemplo, cuando un fragmento de vídeo se codifica como un fragmento intracodificado (I), la unidad de intrapredicción 74 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual, basándose en una modalidad de intrapredicción señalizada y datos de bloques previamente decodificados de la imagen actual. Cuando la imagen se codifica como un fragmento intercodificado (es decir, B, P o GPB), la unidad de compensación de movimiento 72 genera bloques predictivos (también mencionados como muestras de referencia) para un bloque de vídeo del fragmento de vídeo actual, basándose en los vectores de movimiento y en otros elementos sintácticos recibidos desde la unidad de decodificación por entropía 70. Los bloques predictivos se pueden generar a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador de vídeo 30 puede construir las listas de imágenes de referencia, la Lista 0 y la Lista 1, usando técnicas de construcción por omisión, basándose en las imágenes de referencia almacenadas en la memoria de imágenes de referencia 82.

**[0109]** La unidad de decodificación por entropía 70 puede decodificar el flujo de bits utilizando el mismo proceso implementado en el codificador de vídeo 20 (por ejemplo, CABAC, CAVLC, etc.). El proceso de codificación por entropía usado por el codificador se puede señalar en el flujo de bits codificado o puede ser un proceso predeterminado. Por ejemplo, la unidad de decodificación por entropía 70 puede recibir elementos sintácticos binarizados codificados. La unidad de decodificación por entropía 70 puede decodificar el flujo de bits (por ejemplo, usando una modalidad adaptativa al contexto o una modalidad de omisión) y binarizar los valores decodificados para producir elementos sintácticos decodificados.

**[0110]** En algunos casos, la unidad de decodificación por entropía 70 puede decodificar por entropía los datos de predicción. Como se ha indicado anteriormente con respecto al codificador de vídeo 20, los datos de predicción pueden incluir datos que indiquen valores de índices de referencia, vectores de movimiento, predictores de vectores de movimiento, valores de diferencia de vectores de movimiento y similares. Es decir, la unidad de decodificación por entropía 70 puede decodificar por entropía elementos sintácticos que representan vectores de movimiento (por ejemplo, un componente horizontal y componente vertical de los vectores de movimiento), índices de imágenes de referencia y direcciones de predicción. Estos elementos sintácticos pueden proporcionarse para una PU intercodificada, para permitir que el decodificador de vídeo 30 decodifique y reproduzca datos de vídeo definidos por una PU.

**[0111]** En algunos ejemplos, como se ha señalado anteriormente, la unidad de decodificación por entropía 70 puede decodificar por entropía los valores de índices de referencia utilizando tres contextos diferentes (por ejemplo,

ctx0, ctx1 y ctx2). Por ejemplo, la unidad de decodificación por entropía 56 puede decodificar por entropía un primer binario (bin0) y un segundo binario (bin1) usando ctx0 y ctx1, respectivamente, y decodificar un tercer binario (bin2) y otros binarios con contexto ctx2. Compartir el contexto ctx2 entre binarios puede ser ineficaz, debido a las actualizaciones de probabilidad asociadas a la codificación por contexto.

5

**[0112]** De acuerdo a los aspectos de esta divulgación, la unidad de decodificación por entropía 70 puede codificar por CABAC un valor de índice de referencia dedicando ctx2 a codificar bin2 y codificando todos los binarios después de bin2 utilizando una modalidad de codificación por omisión. En otro ejemplo, la unidad de decodificación por entropía 70 puede codificar por contexto menos binarios de un valor de índice de referencia mediante la eliminación de ctx2. Es decir, según los aspectos de esta divulgación, la unidad de decodificación por entropía 70 puede decodificar el bin2 y todos los binarios posteriores utilizando una modalidad de omisión de CABAC. En otro ejemplo más, la unidad de decodificación por entropía 70 puede codificar menos binarios de un valor de índice de referencia mediante la eliminación tanto de ctx1 como de ctx2. Es decir, según aspectos de esta divulgación, la unidad de decodificación por entropía 70 puede decodificar el bin1 y todos los binarios posteriores utilizando una modalidad de omisión de CABAC, reduciendo por ello aún más la complejidad asociada a la codificación de los valores de índices de referencia.

10

15

**[0113]** Otros aspectos de esta divulgación generalmente se refieren a la manera en que la unidad de decodificación por entropía 70 binariza los valores de índices de referencia. En algunos ejemplos, la unidad de decodificación por entropía 70 puede binarizar los valores de índices de referencia utilizando un proceso de codificación unario, unario truncado u otro. En otro ejemplo, la unidad de codificación por entropía 56 puede usar un proceso de codificación de Golomb exponencial para binarizar el valor del índice de referencia.

20

**[0114]** En algunos ejemplos, según aspectos de esta divulgación, la unidad de decodificación por entropía 70 puede implementar una combinación de procesos de binarización. Por ejemplo, como se describe con mayor detalle con respecto a la figura 4 a continuación, la unidad de decodificación por entropía 70 puede combinar un proceso de codificación unario (o truncado unario) con un proceso de codificación de Golomb exponencial para binarizar los valores de índices de referencia. En un ejemplo con fines ilustrativos, la unidad de decodificación por entropía 70 puede combinar un código unario truncado de longitud (4) con un código de Golomb exponencial (por ejemplo, un código de Golomb exponencial de orden 0). En tal ejemplo, la unidad de decodificación por entropía 70 puede binarizar un primer número de binarios (por ejemplo, los dos, tres, cuatro o similares) de un valor de índice de referencia utilizando un código unario, y puede binarizar los binarios restantes del índice de referencia utilizando un código de Golomb exponencial.

25

30

**[0115]** En cualquier caso, la unidad de decodificación por entropía 70 puede implementar las técnicas para la codificación por contexto de uno o más binarios de un valor de índice de referencia y la codificación por omisión de otros uno o más binarios del valor de índice de referencia con cualquier esquema de binarización. Por ejemplo, como se ha señalado anteriormente, la unidad de decodificación por entropía 70 puede codificar por contexto (por ejemplo, codificar por CABAC) un primer número de binarios de un elemento sintáctico binarizado y codificar por omisión los binarios restantes. En el ejemplo descrito anteriormente, en el que un código unario truncado de longitud (4) se combina con un código Golomb exponencial de orden 0, la unidad de decodificación por entropía 70 puede codificar por contexto los dos primeros binarios (o cualquier otro número de binarios) del código unario truncado y luego codificar por omisión la segunda parte del código unario y el código Golomb exponencial completo. En otros ejemplos, la unidad de decodificación por entropía 70 puede usar otros esquemas de binarización. Por ejemplo, la unidad de decodificación por entropía 70 puede usar un código binario de longitud fija en lugar del código de Golomb exponencial descrito en los ejemplos anteriores.

35

40

45

**[0116]** En algunos ejemplos, la unidad de decodificación por entropía 70 puede truncar, o eliminar, los binarios de un valor de índice de referencia binarizado antes de codificar el valor. Adicional o alternativamente, la unidad de decodificación por entropía 70 puede agrupar binarios codificados utilizando el contexto y binarios codificados utilizando una modalidad de omisión. Por ejemplo, la unidad de decodificación por entropía 70 puede codificar índices de referencia de una imagen B mediante codificación por contexto de uno o más binarios de un primer valor de índice de referencia, codificar por contexto uno o más binarios de un segundo valor de índice, codificar por omisión uno o más binarios del primer valor de índice de referencia, y codificar por omisión otros uno o más binarios del segundo valor de índice de referencia (en el orden presentado anteriormente). Por consiguiente, la unidad de decodificación por entropía 70 solo realiza transiciones entre una modalidad de codificación por contexto y una modalidad de codificación por omisión una sola vez, por ejemplo, entre los binarios codificados por contexto y los binarios codificados sin contexto.

50

55

**[0117]** Después de decodificar por entropía los elementos sintácticos y los coeficientes de transformación, en algunos ejemplos, la unidad de decodificación por entropía 70 (o la unidad de cuantización inversa 76) puede recorrer los valores de coeficientes de transformación recibidos, utilizando un recorrido que refleje la modalidad de recorrido utilizado por la unidad de codificación por entropía 56 (o la unidad de cuantización 54) del codificador de vídeo 20. Aunque se muestran como unidades funcionales independientes para facilitar la ilustración, la estructura y la funcionalidad de la unidad de decodificación por entropía 70, la unidad de cuantización inversa 76 y otras unidades del decodificador de vídeo 30 pueden estar sumamente integradas entre sí.

60

65

**[0118]** La unidad de cuantización inversa 76 realiza la cuantización inversa, es decir, decuantiza, los coeficientes de transformación cuantizados, proporcionados en el flujo de bits y decodificados por la unidad de decodificación por entropía 70. El proceso de cuantización inversa puede incluir un proceso convencional, por ejemplo, similar a los procesos propuestos para la HEVC o definidos por la norma de decodificación H.264. El proceso de cuantización inversa también puede incluir el uso de un parámetro de cuantización, QP, calculado por un codificador de vídeo 20 para la CU, para determinar un grado de cuantización y, asimismo, el grado de cuantización inversa que debería aplicarse. La unidad de cuantización inversa 76 puede cuantizar de forma inversa los coeficientes de transformación, ya sea antes o después de que los coeficientes se conviertan, a partir de una matriz unidimensional, en una matriz bidimensional.

**[0119]** La unidad de intrapredicción 74 puede generar datos de predicción para un bloque actual de una trama actual basándose en una modalidad de intrapredicción señalizada y en datos de bloques previamente decodificados de la imagen actual. La unidad de compensación de movimiento 72 puede recuperar el vector de movimiento, la dirección de predicción de movimiento y el índice de referencia a partir del flujo de bits codificado. La dirección de predicción de referencia indica si la modalidad de interpredicción es unidireccional (por ejemplo, una trama P) o bidireccional (una trama B). El índice de referencia indica la imagen de referencia a la que se dirige el vector de movimiento. Basándose en la dirección de predicción de movimiento recuperada, el índice de imagen de referencia y el vector de movimiento, la unidad de compensación de movimiento produce un bloque de movimiento compensado para la parte actual. Estos bloques compensados en movimiento se usan para recrear el bloque predictivo usado para producir los datos residuales.

**[0120]** La unidad de compensación de movimiento 72 puede generar los bloques compensados en movimiento, realizando posiblemente una interpolación basándose en filtros de interpolación. Los identificadores para filtros de interpolación a usar para la estimación de movimiento con una precisión de subpíxel pueden incluirse en los elementos sintácticos. La unidad de compensación de movimiento 72 puede usar filtros de interpolación como los usados por el codificador de vídeo 20 durante la codificación del bloque de vídeo, para calcular valores interpolados para píxeles fraccionarios de un bloque de referencia. La unidad de compensación de movimiento 72 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 de acuerdo a la información sintáctica recibida y usar los filtros de interpolación para generar bloques predictivos.

**[0121]** La unidad de compensación de movimiento 72 puede recibir datos de predicción que indican dónde recuperar información de movimiento para un bloque actual. Por ejemplo, la unidad de compensación de movimiento 72 puede recibir información de predicción de vectores de movimiento, tal como un índice de MVP (`mvp_flag`), una MVD, un indicador de combinación (`merge_flag`) y/o un índice de combinación (`merge_idx`) y usar dicha información para identificar la información de movimiento utilizada para predecir un bloque actual.

**[0122]** Por ejemplo, la unidad de compensación de movimiento 72 puede generar una lista de MVP o candidatos a combinaciones. La unidad de compensación de movimiento 72 puede usar luego un MVP o un índice de combinación para identificar la información de movimiento utilizada para predecir el vector de movimiento de un bloque actual. Es decir, la unidad de compensación de movimiento 72 puede identificar un MVP a partir de una lista de imágenes de referencia utilizando el índice de MVP (`mvp_flag`). La unidad de compensación de movimiento 72 puede combinar el MVP identificado con una MVD recibida para determinar el vector de movimiento para el bloque actual. En otros ejemplos, la unidad de compensación de movimiento 72 puede identificar un candidato a combinación a partir de una lista de imágenes de referencia, utilizando un índice de combinación (`merge_idx`) para determinar la información de movimiento para el bloque actual. En cualquier caso, después de determinar la información de movimiento para el bloque actual, la unidad de compensación de movimiento 72 puede generar el bloque predictivo para el bloque actual.

**[0123]** Adicionalmente, la unidad de compensación de movimiento 72 y la unidad de intrapredicción 74, en un ejemplo de la HEVC, pueden usar parte de la información sintáctica (por ejemplo, proporcionada por un árbol cuádruple) para determinar tamaños de las LCU usadas para codificar una o más imágenes de la secuencia de vídeo codificado. La unidad de compensación de movimiento 72 y la unidad de intrapredicción 74 también pueden usar la información sintáctica para determinar la información de división que describe cómo se divide cada CU de una imagen de la secuencia de vídeo codificado (y, del mismo modo, cómo se dividen las sub-CU). La información sintáctica también puede incluir modalidades que indiquen cómo se codifica cada división (por ejemplo, intrapredicción o interpredicción y, para la intrapredicción, una modalidad de codificación de intrapredicción), una o más imágenes de referencia (y/o listas de referencia que contengan identificadores para las tramas de referencia) para cada PU intercodificada, y otra información para decodificar la secuencia de vídeo codificado.

**[0124]** El sumador 80 suma los bloques residuales con los bloques de predicción correspondientes generados por la unidad de compensación de movimiento 72 o por la unidad de intrapredicción 74, para formar bloques decodificados. Si se desea, también se puede aplicar un filtro de eliminación de bloques para filtrar los bloques decodificados, a fin de eliminar distorsiones de efecto pixelado. Los bloques de vídeo decodificados se almacenan a continuación en la memoria de imágenes de referencia 82, que proporciona bloques de referencia para una posterior compensación de movimiento y también genera vídeo decodificado para su presentación en un dispositivo

de visualización (tal como el dispositivo de visualización 32 de la figura 1).

5 **[0125]** La figura 4 es un diagrama de bloques que ilustra un proceso ejemplar de codificación aritmética. El proceso ejemplar de codificación aritmética de la figura 4 se describe generalmente como realizado por el codificador de vídeo 20. Sin embargo, debería entenderse que las técnicas descritas con respecto a la figura 4 pueden ser realizadas por otros varios codificadores de vídeo, incluyendo el decodificador de vídeo 30. Por ejemplo, como se ha indicado anteriormente con respecto a la figura 3, el decodificador de vídeo 30 puede realizar un proceso de decodificación que es recíproco al proceso realizado por el codificador de vídeo 20.

10 **[0126]** El ejemplo de la figura 4 incluye un binarizador 100, un modelador de contexto 102, un motor de codificación 104 y un codificador por omisión 106. El binarizador 100 es responsable de binarizar un elemento sintáctico recibido. Por ejemplo, el binarizador 100 puede correlacionar un elemento sintáctico con una serie de los llamados binarios, cada uno de los cuales representa un valor binario. En un ejemplo con fines de ilustración, el binarizador 100 puede correlacionar un elemento sintáctico con binarios utilizando el código unario truncado (TU).  
 15 En general, la codificación unaria puede implicar la generación de una cadena de binarios de longitud  $N + 1$ , donde los primeros  $N$  binarios son 1 y los últimos binarios son 0. La codificación unaria truncada puede tener un binario menos que la codificación unaria, al establecer un máximo sobre el valor más grande posible del elemento sintáctico ( $c_{Max}$ ). Un ejemplo de codificación unaria truncada se muestra en la Tabla 2 con  $c_{Max} = 10$ .

20 **TABLA 2**

Valor	Cadena de binarios
0	0
1	10
2	110
3	1110
4	11110
5	111110
6	1111110
7	11111110
8	111111110
9	1111111110
10	1111111111

25 Cuando se realiza en el decodificador de vídeo 30, el decodificador de vídeo 30 puede buscar un 0 para determinar cuándo se completa el elemento sintáctico que se está codificando actualmente. Como se describe con mayor detalle a continuación, la codificación unaria truncada es simplemente un ejemplo, y el binarizador 100 puede realizar otros varios procesos de binarización (así como combinaciones de procesos de binarización) para binarizar elementos sintácticos.

30 **[0127]** El modelador de contexto 102 puede ser responsable de determinar un modelo de contexto (también mencionado como modelo de probabilidad) para un binario dado. Por ejemplo, el modelador de contexto 102 puede seleccionar un modelo de probabilidad que opera sobre el contexto para codificar símbolos asociados a un bloque de datos de vídeo. En general, el modelo de probabilidad almacena la probabilidad de que cada binario sea "1" o "0".

35 **[0128]** El modelador de contexto 102 puede seleccionar el modelo de probabilidad entre una serie de modelos de probabilidad disponibles. En algunos ejemplos, como se describe con mayor detalle a continuación, un contexto utilizado por el modelador de contexto 102 puede determinarse basándose en el número de binario que se está codificando. Es decir, el contexto puede depender de la posición del binario en la cadena de binarios generada por el binarizador 100. En cualquier caso, en el codificador de vídeo 20, un símbolo de destino puede codificarse utilizando el modelo de probabilidad seleccionado. En el decodificador de vídeo 30, se puede analizar sintácticamente un símbolo de destino utilizando el modelo de probabilidad seleccionado.  
 40

45 **[0129]** El motor de codificación 104 codifica el binario utilizando el modelo de probabilidad determinado (procedente del modelador de contexto 102). Después de que el motor de codificación 104 codifique el binario, el motor de codificación 104 puede actualizar el modelo de probabilidad asociado al contexto utilizado para codificar el binario. Es decir, el modelo de probabilidad seleccionado se actualiza en función del valor codificado real (por ejemplo, si el valor del binario fue "1", se aumenta el recuento de frecuencia de un "1"). La codificación de binarios usando el modelador de contexto 102 y el motor de codificación 104 puede denominarse codificación de los binarios utilizando una modalidad de codificación por contexto.

5 **[0130]** El codificador por omisión 106 codifica los binarios utilizando una probabilidad fija. A diferencia de la codificación por contexto (mediante el modelador de contexto 102 y el motor de codificación 104), el codificador por omisión 106 no actualiza el proceso de codificación por omisión basándose en los valores reales de los binarios que se están codificando. Por consiguiente, en general, el codificador por omisión 106 puede codificar por omisión los binarios de código más rápidamente que la codificación por contexto. La codificación de binarios utilizando el codificador por omisión 106 se puede denominar codificación de los binarios usando una modalidad de codificación por omisión. Las modalidades ejemplares de codificación por omisión incluyen la codificación de Golomb, la codificación de Golomb exponencial, la codificación de Golomb-Rice o cualquier otro proceso de codificación adecuado que omita el modelador de contexto 102 y el motor de codificación 104.

15 **[0131]** Los binarios codificados (procedentes del motor de codificación 104 y del codificador por omisión 106) se combinan para formar un flujo de bits codificado. Para decodificar el flujo de bits codificado, un decodificador de vídeo (tal como el decodificador de vídeo 30) puede reflejar el proceso que se muestra en la figura 4. Es decir, el decodificador de vídeo 30 puede realizar la codificación por contexto (usando el modelador de contexto 102 y el motor de codificación 104) o la codificación por omisión (usando el codificador por omisión 106) sobre un flujo de bits codificado para generar una cadena de binarios decodificados. El decodificador de vídeo 30 puede binarizar luego la cadena de binarios (utilizando el binarizador 100) para generar valores sintácticos.

20 **[0132]** El proceso de codificación aritmética mostrado en la figura 3 se puede usar para codificar datos de vídeo. Por ejemplo, el proceso de codificación mostrado en la figura 3 se puede usar para codificar datos de predicción, incluidos valores de índices de referencia, vectores de movimiento, predictores de vectores de movimiento, valores de diferencia de vectores de movimiento y similares.

25 **[0133]** En un ejemplo con fines de ilustración, el binarizador 100 puede binarizar un índice de referencia (ref idx). En algunos ejemplos, la cadena de binarios resultante para el índice de referencia puede tener hasta 15 binarios de longitud, en función del número de imágenes de referencia que estén disponibles para referencia. Como se ha señalado anteriormente, en algunos ejemplos, todos los binarios de un valor de índice de referencia pueden codificarse por contexto utilizando el modelador de contexto 102 y el motor de codificación 104. Además, uno o más de los binarios pueden compartir un contexto. Sin embargo, la codificación por contexto de todos los binarios y la compartición de un contexto entre más de un binario pueden ser ineficaces, debido a las latencias asociadas a la codificación por contexto.

35 **[0134]** De acuerdo a los aspectos de esta divulgación, como se muestra en la figura 4, el codificador de vídeo 20 puede codificar una cadena de binarios para un índice de referencia basándose en el número de binario del binario que se está codificando. Por ejemplo, el codificador de vídeo 20 puede codificar un binario particular en una cadena de binarios de índices de referencia, de acuerdo a una posición relativa del binario particular en la cadena de binarios. En un ejemplo, el codificador de vídeo 20 puede codificar por contexto un primer binario, un segundo binario y un tercer binario de un índice de referencia, y puede codificar por omisión los binarios restantes del índice de referencia. Es decir, el codificador de vídeo 20 puede codificar un primer binario (bin0) usando el modelador de contexto 102 y el motor de codificación 104 con un primer contexto ctx0, un segundo binario (bin1) usando un modelador de contexto 102 y un motor de codificación 104 con un segundo contexto ctx1, y un tercer binario (bin2) usando el modelador de contexto 102 y el motor de codificación 104 con un tercer contexto ctx2. Sin embargo, el codificador de vídeo 20 puede codificar el cuarto binario (bin3) y otros binarios siguientes cualesquiera, utilizando el codificador por omisión 106.

45 **[0135]** En otro ejemplo, el codificador de vídeo 20 puede reducir el número de binarios que están codificados por contexto. Por ejemplo, el codificador de vídeo 20 puede codificar un primer binario (bin0) usando el modelador de contexto 102 y el motor de codificación 104 con un primer contexto ctx0, y un segundo binario (bin1) usando el modelador de contexto 102 y el motor de codificación 104 con un segundo contexto ctx1. Sin embargo, en este ejemplo, el codificador de vídeo 20 puede codificar por omisión un tercer binario (bin2) y otros binarios siguientes cualesquiera, utilizando el codificador por omisión 106.

55 **[0136]** En otro ejemplo más, el codificador de vídeo 20 puede reducir aún más el número de binarios que están codificados por contexto. Por ejemplo, el codificador de vídeo 20 puede codificar un primer binario (bin0) utilizando el modelador de contexto 102 y el motor de codificación 104 con un primer contexto ctx0. Sin embargo, el codificador de vídeo 20 puede codificar el segundo binario (binario 1) y otros binarios siguientes cualesquiera, utilizando el codificador por omisión 106.

60 **[0137]** Aspectos de esta divulgación también se refieren a la manera en que el binarizador 100 realiza la binarización para datos de vídeo. Por ejemplo, según los aspectos de esta divulgación, el binarizador 100 puede dividir un elemento sintáctico en más de una parte. Es decir, el binarizador 100 puede usar la codificación unaria truncada para codificar un prefijo (con un cMax relativamente pequeño, como se ha descrito anteriormente) y puede usar otro procedimiento de codificación para codificar el sufijo. En un ejemplo, el binarizador 100 puede usar un código de Golomb exponencial de orden k para codificar el sufijo.

65

5 **[0138]** En algunos ejemplos, solo los binarios del prefijo pueden estar codificados por contexto, mientras que los binarios del sufijo pueden estar codificados por omisión. La Tabla 3 muestra un ejemplo de código unario truncado combinado con un código exponencial-Golomb, con  $cMax = 4$  para el prefijo y Exp-Golomb de orden 0<sup>ésimo</sup> para el sufijo. Estas técnicas también pueden aplicarse a valores de índices de referencia, así como a otros elementos sintácticos, tales como los valores de diferencia de vectores de movimiento u otros elementos sintácticos utilizados en la codificación, mediante la predicción avanzada de vectores de movimiento (AMVP).

**TABLA 3**

Valor	Cadena de binarios (truncada unaria)	Cadena de binarios (Exp-Golomb de orden 0 <sup>ésimo</sup> )
0	0	
1	10	
2	110	
3	1110	
4	1111	0
5	1111	100
6	1111	101
7	1111	11000
8	1111	11001
9	1111	11010
10	1111	11011

10 En el ejemplo que se muestra en la Tabla 3, los binarios unarios truncados pueden estar codificados por contexto, mientras que los binarios Golomb exponenciales pueden estar codificados por omisión.

15 **[0139]** Las técnicas de esta divulgación pueden incluir, por ejemplo, aplicar la codificación por contexto a los binarios después de un cierto número de binarios para la binarización de Golomb exponencial en una parte de prefijo. Las técnicas de esta divulgación también incluyen, por ejemplo, aplicar la codificación basada en contexto a un cierto número de binarios (por ejemplo, un número predeterminado de binarios de prefijo) y aplicar la codificación por omisión a los binarios restantes. Por ejemplo, en lugar de codificar todos los binarios en una parte de prefijo utilizando el contexto, bin2 y los binarios posteriores en la parte de prefijo pueden codificarse con la modalidad de omisión. En otro ejemplo, la modalidad de omisión se puede aplicar a todos los binarios después de, y/o incluyendo a, bin1. En otro ejemplo más, la modalidad de omisión se puede aplicar a todos los binarios de la parte del prefijo. Para cualquier procedimiento de binarización se puede utilizar un enfoque similar que utiliza la codificación en modalidad de omisión después de un cierto número de binarios codificados por contexto. Es decir, si bien esta divulgación describe el uso de esquemas de codificación exponenciales-Golomb y truncados unarios, se pueden usar otros procedimientos de binarización.

25 **[0140]** En otro ejemplo más, las técnicas de esta divulgación descritas anteriormente pueden implementarse junto con otros procesos de binarización, incluidas combinaciones de procesos de binarización. Es decir, en un ejemplo, se puede usar un proceso de codificación unario para binarizar el valor del índice de referencia. En otro ejemplo, se puede usar un proceso de codificación unario truncado para binarizar el valor del índice de referencia. En otro ejemplo más, se puede usar un proceso de codificación exponencial-Golomb para binarizar el valor del índice de referencia. También son posibles otros procesos de binarización y combinaciones de procesos de binarización. Es decir, por ejemplo, un proceso de codificación unario (o truncado unario) se puede combinar con un proceso de codificación exponencial-Golomb para binarizar el valor del índice de referencia. En un ejemplo con fines de ilustración, un código unario truncado de longitud (4) se puede combinar con un código exponencial-Golomb (por ejemplo, un código exponencial-Golomb de orden 0). En tal ejemplo, un primer número de binarios (por ejemplo, los dos, tres, cuatro o similares) del valor del índice de referencia pueden codificarse de forma unaria, mientras que los binarios restantes del índice de referencia pueden ser codificados con exponencial-Golomb.

30 **[0141]** En cualquier caso, las técnicas descritas anteriormente con respecto a la CABAC y a la codificación por omisión del valor del índice de referencia se pueden aplicar a cualquier valor de índice de referencia binarizado. Es decir, según los aspectos de esta divulgación, un primer número de binarios de un valor de índice de referencia binarizado puede codificarse por contexto (por ejemplo, codificarse con un motor de CABAC), mientras que los binarios restantes pueden codificarse por omisión. En el ejemplo descrito anteriormente, en el que un código unario truncado de la longitud (4) se combina con un código Golomb exponencial de orden 0, los dos primeros binarios (o cualquier otro número de binarios) del código unario truncado pueden ser codificados por contexto, y la segunda parte del código unario y todo el código Golomb exponencial completo pueden ser codificados por omisión.

45 **[0142]** Debería entenderse que el código unario truncado de longitud (4) y el Golomb exponencial de orden 0 se

proporcionan solamente con fines ejemplares, y que se pueden aplicar las técnicas iguales o similares para otras longitudes de código unario truncado, así como para otras órdenes de código Golomb exponencial. Además, los procesos de binarización descritos anteriormente se proporcionan solo a modo de ejemplo, y se pueden usar otros códigos binarizados. Por ejemplo, se puede usar un código binario de longitud fija en lugar del código de Golomb exponencial descrito en los ejemplos anteriores. Además, el ejemplo de dos binarios codificados por contexto para la parte de binarización unitaria truncada se proporciona con fines ilustrativos y se pueden usar otras cantidades de binarios codificados por contexto y por omisión.

**[0143]** En cualquier caso, los aspectos de esta divulgación también se refieren al truncamiento de una parte del valor binarizado. Por ejemplo, debido a que el número de índices de referencia se conoce de antemano, de acuerdo a aspectos de esta divulgación, el código Golomb exponencial o el código de longitud fija se puede truncar. Es decir, el binarizador 100 puede utilizar un Golomb exponencial de orden k. Como ejemplo, un Golomb exponencial de orden 0 puede aplicarse en la compresión de vídeo. Esta binarización consiste en un prefijo exponencial codificado con código unario y sufijo de longitud fija, (*prefijo-1*), un ejemplo del cual se muestra en la Tabla 4 a continuación:

**TABLA 4**

Elemento	Golomb exponencial de orden 0	Código unario	Número binario
0	1	1	000
1	01 0	01	001
2	01 1	001	010
3	001 00	0001	011
4	001 01	00001	100
5	001 10	000001	101
6	001 11	0000001	110
7	0001 000	00000001	111
8	0001 001	000000001	...
9	0001 010	...	
10	0001 011		
11	0001 100		
12	0001 101		
13	0001 110		
14	0001 111		
15	00001 0000		
...	...		

**[0144]** Por ejemplo, el valor 10 (por ejemplo, que corresponde al elemento con el valor 10 en la primera columna de la Tabla 4) está representado por la palabra de código binarizada 0001011, donde 0001 es un prefijo y 011 es un sufijo. El elemento puede ser datos de entrada, que se codifican utilizando las palabras de código de la Tabla 4 o las tablas que se muestran y describen a continuación. Por ejemplo, el codificador de vídeo 20 puede recibir los elementos y convertir los elementos en palabras de código de acuerdo a las tablas que se muestran y describen a continuación. Del mismo modo, el decodificador de vídeo 30 puede recibir las palabras de código y convertir las palabras de código en elementos sintácticos (por ejemplo, datos de entrada) de acuerdo a las tablas que se muestran y describen a continuación.

**[0145]** Un valor reconstruido se puede obtener de acuerdo a la ecuación (1), que se muestra a continuación:

$$\text{valor} = 2^{(\text{prefijo} - 1)} + \text{sufijo} - 1 \quad (1)$$

En este ejemplo, el prefijo está representado por un código unario donde 0001 corresponde a 4, y el sufijo es un valor representado con un sistema de numeración binario donde 011 corresponde a 3, como se muestra en la Tabla 4 anterior. En consecuencia, en este ejemplo, la aplicación de la ecuación (1) da como resultado los siguientes valores:  $2^{(4-1)} + 3 - 1 = 10$ .

**[0146]** Este código puede representar generalmente números infinitos; sin embargo, en algunos escenarios, el número de elementos puede ser conocido. En este caso, la palabra de código se puede acortar, teniendo en cuenta el número máximo de elementos posibles.

**[0147]** Por ejemplo, si el número máximo de elementos es dos (por ejemplo, los elementos 0 y 1), una palabra

normal de código exponencial de Golomb para 1 es 010. Sin embargo, no hay ningún elemento mayor que dos. En consecuencia, el código normal 010 se puede acortar a 1. Este tipo de binarización puede denominarse Golomb Exponencial Truncada y puede usarse en normas de codificación de vídeo tales como la H.264/AVC. Sin embargo, con respecto a la norma H.264, el Golomb exponencial truncado solo se usa cuando el número máximo de elementos es 1. Para otros casos, se utiliza la binarización exponencial normal de Golomb.

**[0148]** De acuerdo a aspectos de esta divulgación, la codificación de Golomb exponencial normal puede truncarse adicionalmente, por ejemplo, de una manera similar a la del ejemplo anterior, en el que el número máximo de elementos es 1. Generalmente, cuando se conoce de antemano el número máximo de elementos, el sufijo de la palabra de código de binarización de Golomb exponencial se puede truncar eliminando los binarios redundantes. Por ejemplo, si el número máximo de elementos es 9, entonces los dos binarios marcados con negrita, cursiva y subrayado en la Tabla 5, que se muestra a continuación, se pueden eliminar de las palabras de código.

**TABLA 5**

Elemento	Golomb exponencial de orden 0	Golomb Exponencial Truncado de orden 0
0	1	1
1	01 0	01 0
2	01 1	01 1
3	001 00	001 00
4	001 01	001 01
5	001 10	001 10
6	001 11	001 11
7	0001 <i><u>00</u></i>	0001 0
8	0001 <i><u>00</u></i> 1	01

Es decir, para el elemento 7 que se muestra en la Tabla 5, se pueden eliminar los dos primeros 00 del sufijo. Además, para el elemento 8 que se muestra en la Tabla 5, se pueden eliminar los dos primeros 00 del sufijo. Por consiguiente, el Golomb Exponencial Truncado de orden 0 indica el 0001 0 para el elemento 7 y el 00001 1 para el elemento 8.

**[0149]** Las técnicas descritas anteriormente se pueden implementar, por ejemplo, comparando un sufijo de longitud fija para el prefijo más reciente (0001 en el ejemplo anterior) con el código normal exponencial de Golomb. Por ejemplo, si el número de elementos en el grupo más reciente es más pequeño que en un código de Golomb Exponencial normal, se pueden eliminar los binarios redundantes. En otras palabras, el binarizador 100 puede generar el código Golomb exponencial truncado resultante de orden 0, al comparar el sufijo de longitud fija para el prefijo más reciente y, si el número de elementos en este grupo más reciente es más pequeño que en el código Golomb Exponencial normal, los binarios redundantes se puede quitar.

**[0150]** Por ejemplo, en este ejemplo, el binarizador 100 puede determinar el número de elementos cuyo prefijo es el mismo que el prefijo del último elemento cuando hay un número máximo de elementos a codificar que se conoce de antemano. Por ejemplo, en la Tabla 5, el prefijo para el último elemento es 0001, y hay dos elementos (por ejemplo, el elemento 7 y el elemento 8) cuyo prefijo es el mismo que el prefijo del último elemento cuando hay un número máximo de elementos (por ejemplo, 9 en este ejemplo).

**[0151]** El binarizador puede entonces comparar el número de elementos cuyo prefijo es el mismo que el prefijo del último elemento con el número de elementos en el código de Golomb Exponencial normal con el mismo prefijo. Por ejemplo, en la Tabla 4 anterior, hay ocho elementos (es decir, el elemento 7 al elemento 14) cuyo prefijo es 0001 (es decir, el mismo que el prefijo del elemento anterior). En este ejemplo, el binarizador 100 puede determinar que el número de elementos cuyo prefijo es el mismo que el prefijo del último elemento es menor que el número de elementos en el código de Golomb Exponencial normal con el mismo prefijo.

**[0152]** Cuando esto es cierto, el binarizador 100 puede truncar binarios de palabras de código cuyo prefijo es el mismo que el último prefijo, para generar palabras de código truncadas. En algunos ejemplos, los binarios se truncan a partir del sufijo; aunque los aspectos de esta divulgación no están limitados de ese modo. El binarizador 100 puede determinar el número de binarios a truncar basándose en el número de elementos cuyo prefijo es el mismo que el último prefijo.

**[0153]** Por ejemplo, en la Tabla 5 anterior, hay dos elementos con el mismo prefijo que el último prefijo (por ejemplo, los elementos 7 y 8). El binarizador 100 puede truncar binarios de palabras de código de los elementos 7 y 8 para generar palabras de código truncadas, como se ilustra en la última columna de la Tabla 5. En este ejemplo, debido a que hay dos elementos con el mismo prefijo que el último prefijo, el binarizador 100 puede determinar que solo se necesita un binario en el sufijo para representar los dos elementos. Por ejemplo, un 0 en el sufijo puede

representar un elemento (por ejemplo, el elemento 7), y un 1 en el sufijo puede representar otro elemento (por ejemplo, el elemento 8). Por consiguiente, para el elemento 7 en la Tabla 5 anterior, el binarizador 100 puede truncar los dos primeros binarios del sufijo, dejando solo 0 como el sufijo para la palabra de código truncada. Además, para el elemento 8 en la Tabla 5 anterior, el binarizador 100 puede truncar los dos primeros binarios del sufijo, dejando solo 1 como el sufijo para la palabra de código truncada.

**[0154]** Las técnicas descritas anteriormente pueden implementarse para codificar medios (tal como codificar y/o decodificar datos de vídeo). Por ejemplo, de acuerdo a aspectos de esta divulgación, un decodificador de vídeo, tal como el decodificador de vídeo 30, puede recibir una o más palabras de código, representativas de los datos de medios, y puede haber un número máximo de elementos que pueden usarse para la codificación. El decodificador de vídeo 30 puede convertir las palabras de código en elementos de acuerdo a una tabla de codificación. La tabla de codificación puede construirse de manera tal que al menos algunas de las palabras de código que tienen el mismo prefijo estén truncadas cuando el mismo prefijo sea el último prefijo en la tabla de codificación, y el número de palabras de código que tienen el mismo prefijo sea menor que un número máximo de palabras de código únicas que podrían tener el mismo prefijo. Por ejemplo, para el prefijo 0001, la Tabla 4 ilustra posibilidades únicas para las palabras de código, y las Tablas 5 y 6 (que se muestran a continuación) muestran ejemplos de palabras de código que comparten el mismo prefijo y se truncan de acuerdo a las técnicas de esta divulgación.

**[0155]** Las técnicas también pueden ser realizadas por el codificador de vídeo 20. Por ejemplo, el codificador de vídeo 20 puede recibir uno o más elementos representativos de datos de medios. El codificador de vídeo 20 puede convertir los elementos en una o más palabras de código de acuerdo a una tabla de codificación, y puede haber un número máximo de elementos que se pueden usar para la codificación. La tabla de codificación puede construirse de manera tal que al menos algunas de las palabras de código que tienen el mismo prefijo estén truncadas cuando el mismo prefijo sea el último prefijo en la tabla de codificación, y el número de palabras de código que tienen el mismo prefijo sea menor que un número máximo de palabras de código únicas que podrían tener el mismo prefijo. Nuevamente, por ejemplo, para el prefijo 0001, la Tabla 4 ilustra posibilidades únicas para palabras de código, y las Tablas 5 y 6 (que se muestran a continuación) muestran ejemplos de palabras de código que comparten el mismo prefijo y se truncan de acuerdo a las técnicas de esta divulgación.

**[0156]** De esta manera, las técnicas pueden reducir el número de binarios necesarios para codificar datos de vídeo cuando se conoce el número máximo de elementos que se están codificando. La reducción en los binarios da como resultado menos bits que necesitan ser señalizados o recibidos, dando como resultado la eficacia del ancho de banda.

**[0157]** En otro ejemplo más, si el número máximo de elementos es 11, las palabras de código de la codificación Golomb Exponencial Truncada se muestran en la Tabla 6 a continuación. Los binarios marcadas con negrita, cursiva y subrayado en la Tabla 6 se pueden eliminar de las palabras de código.

**TABLA 6**

Elemento	Golomb exponencial de orden 0	Golomb Exponencial Truncado de orden 0
0	1	1
1	01 0	01 0
2	01 1	01 1
3	001 00	001 00
4	001 01	001 01
5	001 10	001 10
6	001 11	001 11
7	0001 <b><i><u>0</u></i></b> 00	0001 00
8	0001 <b><i><u>0</u></i></b> 01	0001 01
9	0001 <b><i><u>0</u></i></b> 10	0001 10
10	0001 <b><i><u>0</u></i></b> 11	0001 11

**[0158]** Como se muestra en la Tabla 6, los primeros binarios en el sufijo para los elementos 7, 8, 9 y 10 se pueden truncar (se muestran en negrita, cursiva y subrayado). En este ejemplo, solo se puede truncar un binario del sufijo, porque la palabra de código representa cuatro elementos. Por esta razón, en las palabras de código truncadas, el sufijo comienza desde 00 y termina en 11 para abarcar cuatro elementos, cada uno con el mismo prefijo.

**[0159]** Los ejemplos que se muestran en las Tablas 5 y 6 anteriores se proporcionan solo como ejemplos, y se puede aplicar el mismo proceso para cualquier número máximo de elementos. Por ejemplo, en algunos ejemplos, aspectos de esta divulgación se refieren a recibir palabras de código truncadas. Las palabras de código truncadas

se pueden generar determinando un primer número de elementos. El primer número de elementos puede indicar cuántas palabras de código en una primera tabla de codificación tienen un prefijo que es el mismo que un prefijo de una palabra de código que corresponde a un último elemento en la primera tabla de codificación cuando hay un número máximo de elementos que pueden ser utilizados para la codificación. En este ejemplo, la primera tabla de codificación puede ser la Tabla 5 o la Tabla 6. Los aspectos se refieren a la recepción de las palabras de código truncadas que están preclasificadas o calculadas sobre la marcha durante la operación.

**[0160]** Los aspectos de esta divulgación también se refieren a la determinación de un segundo número de elementos que indican cuántas palabras de código en una segunda tabla de codificación tienen un prefijo que es el mismo que el prefijo de las palabras de código que corresponde al último elemento en la primera tabla de codificación. En este ejemplo, la segunda tabla de codificación puede ser la Tabla 4 anterior. En algunos ejemplos, la primera tabla de codificación puede ser un subconjunto de la segunda tabla de codificación, que se basa en el número máximo de elementos que se pueden usar para la codificación.

**[0161]** En algunos ejemplos, cuando el primer número de elementos es menor que el segundo número de elementos, aspectos de esta divulgación se refieren al truncamiento de binarios de las palabras de código en la primera tabla de codificación cuyo prefijo es el mismo que el prefijo de la palabra de código que corresponde al último elemento en la primera tabla de codificación para generar palabras de código truncadas, y a la codificación de los datos de vídeo usando las palabras de código truncadas. En algunos ejemplos, truncar las palabras de código incluye truncar binarios de sufijos o prefijos de las palabras de código, o una combinación de los mismos. En algunos ejemplos, los intervalos de truncamiento se basan en el primer número de elementos, donde el primer número de elementos indica cuántas palabras de código en una primera tabla de codificación tienen un prefijo que es el mismo que un prefijo de una palabra de código que corresponde a un último elemento en la primera tabla de codificación. En algunos ejemplos, la codificación es la codificación de Golomb.

**[0162]** De forma alternativa o adicional, un prefijo también puede acortarse utilizando un código unario truncado. Por ejemplo, si el número máximo de elementos es 4, entonces se pueden truncar un prefijo y un sufijo como se muestra en la Tabla 7 a continuación.

**TABLA 7**

Elemento	Golomb exponencial de orden 0	Golomb Exponencial Truncado de orden 0
0	1	1
1	01 0	01 0
2	01 1	01 1
3	00 <b><i><u>1 00</u></i></b>	00

**[0163]** Los binarios truncados de la Tabla 7 se representan en negrita, cursiva y subrayado. En el ejemplo que se muestra en la Tabla 7, la palabra de código para el elemento 3 es más corta que las palabras de código para los elementos 1 o 2. Se puede aplicar un reordenamiento o correlación adicional para la binarización de Golomb exponencial truncada, por ejemplo, asignando la palabra de código más corta 00 al elemento 1 que aparece más frecuentemente y 010 al elemento 3. Tal reordenamiento o correlación se puede realizar usando tablas de correlación.

**[0164]** En algunos ejemplos, el reordenamiento también puede basarse en la frecuencia de aparición de elementos particulares. Por ejemplo, las palabras de código más cortas se pueden asignar a los elementos que aparecen con mayor frecuencia. Esta correlación de palabras de código puede ser particularmente eficaz en casos en que los elementos están ordenados por la frecuencia de aparición.

**[0165]** Si bien ciertos ejemplos anteriores se describieron con respecto a la codificación Golomb exponencial de orden 0, debería entenderse que las técnicas son más generalmente aplicables a la codificación Golomb exponencial de orden  $k$ . Además, las técnicas no se limitan a la norma de vídeo HEVC, y se pueden aplicar a cualquier norma de compresión de vídeo o, más ampliamente, para cualquier aplicación en la que se realice la binarización.

**[0166]** Con respecto a la norma HEVC emergente (así como las extensiones a la norma HEVC, tales como la codificación de vídeo ajustable a escala (SVC) o la codificación de vídeo de vistas múltiples (MVC)), las técnicas de binarización de la codificación Golomb exponencial truncada, descritas anteriormente, pueden aplicarse para binarizar varios elementos sintácticos. Los ejemplos incluyen valores de índices de referencia, una modalidad intramodalidad, un índice de combinación, parámetros de cuantización (o parámetros de cuantización delta), o cualquier otro elemento sintáctico para el cual se conoce de antemano una serie de elementos.

**[0167]** Si bien el ejemplo anterior describe el truncamiento exponencial de Golomb, las técnicas de truncamiento descritas también pueden aplicarse a un código de longitud fija. Es decir, en los ejemplos en los que un elemento

sintáctico (por ejemplo, un índice de referencia) se binariza utilizando más de un proceso de binarización (por ejemplo, Golomb truncado unario y exponencial), un número predeterminado de binarios puede codificarse con CABAC, mientras que los binarios restantes pueden ser truncados y codificados por omisión.

5 **[0168]** En algunos ejemplos, se puede aplicar un algoritmo para determinar el número de binarios que pueden truncarse (por ejemplo, truncarse desde la parte de binarización de Golomb exponencial o de longitud fija del elemento sintáctico binarizado). En un ejemplo, supóngase que un número predeterminado de binarios quedan para ser codificados por omisión. En este ejemplo, un codificador de vídeo (tal como el codificador de vídeo 20 o el decodificador de vídeo 30) puede determinar el número de binarios restantes que pueden truncarse, calculando un log2 redondeado al alza de los binarios restantes.

10 **[0169]** Las figuras 5A y 5B son diagramas de bloques que ilustran cadenas de binarios ejemplares, asociadas a datos de predicción. Por ejemplo, la figura 5A generalmente ilustra un índice de referencia (ref\_idx), una diferencia de vectores de movimiento (mvd) y un índice de predictores de vectores de movimiento (mvp\_idx) para una imagen que se predice a partir de una imagen de referencia única.

15 **[0170]** La figura 5B generalmente ilustra un primer índice de referencia (ref\_idx\_L0), una primera diferencia de vectores de movimiento (mvd\_L0) (que representa el componente horizontal y el componente vertical), y un primer índice de predictor de vectores de movimiento (mvp\_idx\_L0), así como un segundo índice de referencia (ref\_idx\_L1), una segunda diferencia de vectores de movimiento (mvd\_L1) (que representa el componente horizontal y el componente vertical), y un segundo índice de predictor de vector de movimiento (mvp\_idx\_L1) para una imagen que se predice a partir de dos imágenes de referencia (una imagen B). Es decir, para una PU bipredicha, se pueden codificar dos índices de referencia, con un índice de referencia para cada lista entre lista L0 y lista L1. Por consiguiente, hasta dos índices de referencia pueden codificarse por cada PU, y hasta ocho índices pueden codificarse por cada CU.

20 **[0171]** Las cadenas de binarios 120 (Figura 5A) y 124 (Figura 5B) incluyen datos de predicción asociados a una técnica de predicción avanzada de vectores de movimiento (AMVP). Con la AMVP, un vector de movimiento para un bloque que se está codificando actualmente puede codificarse como un valor de diferencia (es decir, delta) con respecto a otro vector de movimiento, tal como un vector de movimiento asociado a un bloque espacial o temporalmente vecino. Por ejemplo, el codificador de vídeo 20 puede construir una lista de candidatos de predictores de vectores de movimiento que incluye vectores de movimiento asociados a uno o más bloques vecinos en direcciones espaciales y temporales. El codificador de vídeo 20 puede seleccionar el predictor de vector de movimiento (MVP) más preciso entre la lista de candidatos basándose, por ejemplo, en un análisis de costes de velocidad-distorsión.

25 **[0172]** El codificador de vídeo 20 puede indicar la imagen de referencia para el vector de movimiento real usando el índice de referencia (ref\_idx). Además, el codificador de vídeo 20 puede indicar el MVP seleccionado utilizando el índice de predictor de vectores de movimiento (mvp\_idx), que identifica al MVP en la lista de candidatos. El codificador de vídeo 20 también puede indicar la diferencia entre el vector de movimiento del bloque actual (el vector de movimiento real) y el MVP, utilizando la diferencia de vector de movimiento (mvd). Como se ha indicado anteriormente, el mvd puede incluir un componente horizontal y un componente vertical correspondientes al componente horizontal y al componente vertical del mvp.

30 **[0173]** El decodificador de vídeo 30 puede construir una lista de candidatos a MVP de la misma manera. El decodificador de vídeo 30 puede usar entonces el índice de predictor de vector de movimiento (mvp\_idx) recibido para determinar dónde ubicar el MVP en la lista de candidatos. El decodificador de vídeo 30 puede combinar la diferencia de vector de movimiento (mvd) con el predictor de vector de movimiento (determinado usando el índice del predictor de vector de movimiento (mvp\_idx)) para reconstruir el vector de movimiento.

35 **[0174]** La predicción de los vectores de movimiento de esta manera (por ejemplo, con un valor de diferencia) puede requerir que se incluyan menos bits en el flujo de bits en relación con la codificación de los uno o más vectores de movimiento reales. Con respecto a la figura 5B, las imágenes bipredichas pueden incluir datos de predicción asociados a imágenes de dos listas diferentes, por ejemplo, la lista 0 y la lista 1. Como se muestra en el ejemplo de la figura 5B, los datos de predicción asociados a la lista 0 pueden preceder a los datos de predicción asociados a la lista 1. Es decir, la cadena de binarios 124 incluye un primer índice de referencia (ref\_idx\_L0), una primera diferencia de vectores de movimiento (mvd\_L0) (por ejemplo, que representa ambos componentes horizontal y vertical), y un primer índice de predictor de vectores de movimiento (mvp\_idx\_L0), seguido de un segundo índice de referencia (ref\_idx\_L1), una segunda diferencia de vectores de movimiento (mvd\_L1) (por ejemplo, que representa a ambos componentes horizontal y vertical), y un segundo índice de predictor de vectores de movimiento (mvp\_idx\_L1).

40 **[0175]** En algunos ejemplos, la sintaxis relacionada con la AMVP puede codificarse utilizando una combinación de codificación por contexto y codificación por omisión. Por ejemplo, como se muestra en los ejemplos de las figuras 5A y 5B, algunos de los binarios de datos de predicción están codificados por contexto, mientras que otros binarios están codificados por omisión. Es decir, uno o más binarios de valores de diferencia de vectores de

movimiento (así como otros valores, tales como los valores de índice de referencia, según lo descrito con respecto a la Figura 7 a continuación) pueden estar codificados por contexto, mientras que otros uno o más binarios de valores de diferencia de vectores de movimiento pueden ser codificados por omisión.

5 **[0176]** Con respecto al ejemplo de la figura 5A, el índice de referencia (ref\_idx) y una primera parte de la diferencia de vectores de movimiento (mvd) pueden estar codificados por contexto, según lo indicado por los binarios codificados por contexto 128. Una segunda parte de la diferencia de vectores de movimiento (mvd) puede codificarse por omisión, según lo indicado por los binarios codificados por omisión 130. Además, el índice del predictor del vector de movimiento (mpv\_idx) puede estar codificado por contexto, según lo indicado por los binarios  
10 codificados por contexto 132.

**[0177]** Con respecto al ejemplo de la figura 5B, el primer índice de referencia (ref\_idx\_L0) y una primera parte de la primera diferencia de vectores de movimiento (mvd\_L0) (por ejemplo, que representan ambos componentes horizontal y vertical) pueden estar codificados por contexto, según lo indicado por los binarios codificados por contexto 136. Una segunda parte de la primera diferencia de vectores de movimiento (mvd\_L0) (por ejemplo, que representa ambos componentes horizontales y verticales) puede codificarse por omisión, según lo indicado por los binarios codificados por omisión 138. Además, el primer índice de predicción de vector de movimiento (mpv\_idx\_L0), el segundo índice de referencia (ref\_idx\_L1) y una primera parte de la segunda diferencia de vectores de movimiento (mvd\_L1) (por ejemplo, que representa ambos componentes horizontal y vertical) pueden  
15 codificarse por contexto, según lo indicado por los binarios codificados por contexto 140. Una segunda parte de la segunda diferencia de vectores de movimiento (mvd\_L1) (por ejemplo, que representa ambos componentes horizontales y verticales) también puede codificarse por omisión, según lo indicado por los binarios codificados por omisión 142. Finalmente, el segundo índice de predictor de vector de movimiento (mpv\_idx\_L1) puede estar  
20 codificado por contexto, según lo indicado por los binarios codificados por contexto 144.

**[0178]** En consecuencia, el ejemplo de la figura 5B ilustra la sintaxis de intermodalidad basada en las PU para la predicción doble, por la cual un codificador de vídeo puede tener que conmutar cuatro veces entre la codificación por contexto y la codificación por omisión para procesar los binarios. La conmutación entre la codificación por contexto y la codificación por omisión para codificar las cadenas de binarios 120 y 124 puede ser ineficaz. Por  
25 ejemplo, la conmutación entre la codificación por contexto y la codificación por omisión puede consumir uno o más ciclos de reloj. En consecuencia, la conmutación entre la codificación por contexto y la codificación por omisión para cada elemento puede introducir latencia, debido a las transiciones entre la codificación por contexto y la codificación por omisión.

**[0179]** De acuerdo a aspectos de esta divulgación, como se describe con mayor detalle con respecto a las figuras 6 y 7 a continuación, los binarios por contexto y los binarios por omisión pueden agruparse para reducir las transiciones entre la codificación por contexto y la codificación por omisión. Por ejemplo, con respecto a la figura 5A, los aspectos de esta divulgación se refieren a agrupar entre sí los binarios codificados por contexto 128 y 132, de modo que los binarios no estén separados por los binarios codificados por omisión 130. De esta manera, se  
30 puede hacer una transición única entre la codificación por contexto y la codificación por omisión durante la codificación de la cadena de binarios 120.

**[0180]** Asimismo, con respecto a la figura 5B, aspectos de esta divulgación se refieren a agrupar los binarios codificados por contexto 136, 140 y 144, de modo que los binarios no estén separados por los binarios codificados por omisión 138 y 142. De nuevo, agrupar los binarios codificados por contexto y los binarios codificados por omisión de esta manera puede permitir que un codificador de vídeo (tal como el codificador de vídeo 20 o el decodificador de vídeo 30) realice una única transición entre la codificación por contexto y la codificación por omisión. Evitar múltiples transiciones entre la codificación por contexto y la codificación por omisión puede  
35 aumentar la eficacia al limitar la latencia asociada a las transiciones.

**[0181]** La figura 6 es un diagrama de bloques que ilustra otra cadena ejemplar de binarios 140 de datos de predicción. La cadena de binarios 140 incluye un primer índice de referencia (ref\_idx\_L0), una primera diferencia de vectores de movimiento (mvd\_L0) (por ejemplo, que representa ambos componentes horizontal y vertical), y un primer índice de predictor de vector de movimiento (mpv\_idx\_L0), seguido de un segundo índice de referencia (ref\_idx\_L1), una segunda diferencia de vectores de movimiento (mvd\_L1) (por ejemplo, que representa ambos  
40 componentes horizontal y vertical), y un segundo índice de predictor de vector de movimiento (mpv\_idx\_L1). La cadena de binarios 140 incluye los binarios codificados por contexto 144 y los binarios codificados por omisión 148. Por ejemplo, los binarios codificados por contexto 144 pueden codificarse utilizando una modalidad de codificación por contexto de un proceso de codificación adaptativo al contexto (por ejemplo, CABAC), mientras que los binarios codificados por omisión 148 pueden codificarse utilizando una probabilidad fija (por ejemplo, una modalidad de codificación por omisión de CABAC).  
45

**[0182]** De acuerdo a aspectos de esta divulgación, los binarios codificados por contexto 144 se agrupan para la codificación antes de los binarios por omisión 148. Es decir, en el ejemplo mostrado en la figura 6, los binarios codificados por contexto 144 incluyen los binarios codificados por contexto de un primer índice de referencia (ref\_idx\_L0), los binarios codificados por contexto de un segundo índice de referencia (ref\_idx\_L1), los binarios  
50

55  
60  
65

codificados por contexto de un primer índice predictor de vector de movimiento (mvp\_idx\_L0), los binarios codificados por contexto de un segundo índice predictor de vectores de movimiento (mvp\_idx\_L1), los binarios codificados por contexto de una primera diferencia de vectores de movimiento (mvd\_L0) y los binarios codificados por contexto de una segunda diferencia de vectores de movimiento (mvd\_L1). Además, los binarios codificados por omisión 148 incluyen los binarios codificados por omisión de la primera diferencia de vectores de movimiento (mvd\_L0) y los binarios codificados por omisión de la segunda diferencia de vectores de movimiento (mvd\_L1).

**[0183]** En algunos ejemplos, los intervalos se pueden agrupar en función de los números de binario de los elementos sintácticos que se codifican. En un ejemplo con fines de ilustración, un codificador de vídeo (tal como el codificador de vídeo 20 o el decodificador de vídeo 30) puede codificar los primeros dos binarios de los valores de diferencia de vectores de movimiento utilizando la codificación por contexto y codificar los binarios restantes utilizando la codificación por omisión. En consecuencia, en este ejemplo, los binarios codificados por contexto 144 pueden incluir los dos primeros binarios, tanto de la primera diferencia de vectores de movimiento (mvd\_L0) como del segundo valor de diferencia de vectores de movimiento (mvd\_L1), mientras que los binarios codificados por omisión 148 pueden incluir los binarios restantes de los valores de diferencia de vectores de movimiento.

**[0184]** El ordenamiento de los elementos sintácticos mostrados en la figura 6 se proporciona solamente con fines de ejemplo. En otros ejemplos, los elementos sintácticos pueden ordenarse de manera diferente, por ejemplo, con valores de índice de referencia, predictores de vectores de movimiento y valores de diferencia de vectores de movimiento en un orden alternativo (o intercalados). Es decir, en otro ejemplo, un codificador de vídeo puede codificar una PU de la siguiente manera: binarios codificados por contexto del índice de referencia L0, binarios codificados por contexto del índice de referencia L1, binarios codificados en modalidad de omisión del índice de referencia L0, binarios codificados en modalidad de omisión del índice de referencia L1. En otros ejemplos, los binarios por omisión 148 se pueden codificar antes de los binarios codificados por contexto 144.

**[0185]** En cualquier caso, la figura 6 muestra datos de predicción que tienen un grupo de uno o más binarios codificados por contexto 144 y un grupo de uno o más binarios de codificación por omisión 148. Agrupar los binarios codificados por contexto y los binarios codificados por omisión de esta manera puede, como se ha señalado anteriormente, reducir la latencia asociada a la codificación de los binarios. Por ejemplo, un codificador de vídeo puede codificar todos los binarios codificados por contexto 144, hacer una transición única desde la codificación por contexto a la codificación por omisión y codificar todos los binarios codificados por omisión. Evitar múltiples transiciones entre la codificación por contexto y la codificación por omisión puede aumentar la eficacia al limitar la latencia asociada a la transición entre la codificación por contexto y la codificación por omisión.

**[0186]** En algunos ejemplos, como se muestra en la figura 6, los binarios codificados por contexto y los binarios codificados por omisión se pueden agrupar para un bloque de datos de vídeo (por ejemplo, por cada PU). En otros ejemplos, los binarios codificados por contexto y los binarios codificados por omisión se pueden agrupar para las CU (por ejemplo, una o más CU de una LCU), para una LCU completa o para un fragmento completo de datos de vídeo. En dichos ejemplos, los binarios codificados por contexto pueden agruparse y codificarse para una CU / una LCU / un fragmento antes de los binarios codificados por omisión de la CU / de la LCU / del fragmento, o viceversa.

**[0187]** La figura 7 es un diagrama de bloques que ilustra otra cadena ejemplar de binarios 160 de datos de predicción. En el ejemplo de la figura 7, la cadena de binarios 160 incluye un primer índice de referencia (ref\_idx\_L0), una primera diferencia de vectores de movimiento (mvd\_L0) (por ejemplo, que representa ambos componentes horizontal y vertical), y un primer índice de predictor de vector de movimiento (mvp\_idx\_L0), seguido de un segundo índice de referencia (ref\_idx\_L1), una segunda diferencia de vectores de movimiento (mvd\_L1) (por ejemplo, que representa ambos componentes horizontal y vertical), y un segundo índice de predictor de vector de movimiento (mvp\_idx\_L1). La cadena de binarios 160 incluye los binarios codificados por contexto 164 y los binarios codificados por omisión 168. Por ejemplo, los binarios codificados por contexto 164 pueden codificarse utilizando una modalidad de codificación por contexto de un proceso de codificación adaptativa al contexto (por ejemplo, CABAC), mientras que los binarios codificados por omisión 168 pueden codificarse utilizando una probabilidad fija (por ejemplo, una modalidad de codificación por omisión de CABAC).

**[0188]** De acuerdo a aspectos de esta divulgación, de manera similar al ejemplo mostrado en la figura 6, los binarios codificados por contexto 164 se agrupan para la codificación antes de los binarios por omisión 168. Sin embargo, en el ejemplo mostrado en la figura 7, los índices de referencia (ref\_idx\_L0 y ref\_idx\_L1) incluyen una combinación de binarios codificados por contexto, así como binarios codificados por omisión. Es decir, los índices de referencia pueden codificarse de acuerdo a los ejemplos descritos con respecto a la figura 4 anterior, con uno o más binarios codificados utilizando una modalidad adaptativa al contexto y uno o más binarios codificados utilizando una modalidad de omisión.

**[0189]** En consecuencia, en el ejemplo de la figura 7, los binarios 164 codificados por contexto incluyen los binarios codificados por contexto de un primer índice de referencia (ref\_idx\_L0), los binarios codificados por contexto de un segundo índice de referencia (ref\_idx\_L1), los binarios codificados por contexto de un primer índice de predictor de vectores de movimiento (mvp\_idx\_L0), los binarios codificados por contexto de un segundo índice de predictor de vectores de movimiento (mvp\_idx\_L1), binarios codificados por contexto de una primera diferencia

de vectores de movimiento (mvd\_L0) (por ejemplo, que representan ambos componentes horizontales y verticales) y binarios codificados por contexto de una segunda diferencia de vectores de movimiento (mvd\_L1) (por ejemplo, que representan ambos componentes horizontales y verticales). Además, los binarios codificados por omisión 168 incluyen los binarios codificados por omisión del primer índice de referencia (ref\_idx\_L0), los binarios codificados por omisión del segundo índice de referencia (ref\_idx\_L1), los binarios codificados por omisión de la primera diferencia de vectores de movimiento (mvd\_L0) y los binarios codificados por omisión de la segunda diferencia de vectores de movimiento (mvd\_L1).

**[0190]** Como se ha descrito anteriormente con respecto a la figura 6, los binarios se pueden agrupar en función de los números de binario de los elementos sintácticos que se codifican. En un ejemplo con fines de ilustración, un codificador de vídeo (tal como el codificador de vídeo 20 o el decodificador de vídeo 30) puede codificar los primeros dos binarios de los valores de diferencia de vectores de movimiento utilizando la codificación por contexto y codificar los binarios restantes utilizando la codificación por omisión. Además, el codificador de vídeo puede codificar los dos primeros binarios de los índices de referencia usando la codificación por contexto y codificar los binarios restantes usando la codificación por omisión. En consecuencia, en este ejemplo, los binarios codificados por contexto 144 pueden incluir los dos primeros binarios, tanto de la primera diferencia de vectores de movimiento (mvd\_L0) como del segundo valor de la diferencia de vectores de movimiento (mvd\_L1), y los dos primeros binarios, tanto del primer índice de referencia (ref\_idx\_L0) como del segundo índice de referencia (ref\_idx\_L1). Los binarios codificados por omisión 148 pueden incluir los binarios restantes de los valores de diferencia de vectores de movimiento y los índices de referencia.

**[0191]** Agrupar los binarios codificados por contexto y los binarios codificados por omisión puede reducir la latencia asociada a la codificación de los binarios. Por ejemplo, un codificador de vídeo puede codificar todos los binarios codificados por contexto 144, hacer una transición única desde la codificación por contexto a la codificación por omisión y codificar todos los binarios codificados por omisión. Evitar múltiples transiciones entre la codificación por contexto y la codificación por omisión puede aumentar la eficacia al limitar la latencia asociada a la transición entre la codificación por contexto y la codificación por omisión.

**[0192]** Debería entenderse que las técnicas descritas con respecto a las figuras 6 y 7 pueden realizarse utilizando cualquier esquema de binarización. Además, como se ha indicado anteriormente, el ordenamiento de los elementos sintácticos se proporciona solo con fines ejemplares. Además, los binarios codificados por contexto y los binarios codificados por omisión se pueden agrupar para una PU, para una o más CU, para una LCU completa o para un fragmento completo de datos de vídeo. En dichos ejemplos, los binarios codificados por contexto se pueden agrupar y codificar para una PU / una CU / una LCU / un fragmento antes de los binarios codificados por omisión de la PU / la CU / la LCU / el fragmento, o viceversa.

**[0193]** Las figuras 8A y 8B generalmente ilustran la codificación de un elemento sintáctico de dirección de interpredicción. Por ejemplo, como se ha señalado anteriormente, además de calcular el vector de movimiento, la estimación del movimiento también determina el índice de la trama de referencia (ref\_idx) y la dirección de predicción para los fragmentos B (inter\_pred\_idc: hacia adelante desde L0, hacia atrás desde L1, o bidireccional, o indicador\_inter\_pred: unidirección desde LC o bidirección desde L0 y L1). El vector de movimiento (por ejemplo, un componente horizontal y un componente vertical del vector de movimiento), el índice de trama de referencia y la dirección de predicción son típicamente codificados por entropía por el codificador como elementos sintácticos, y se colocan en el flujo de bits de vídeo codificado para ser decodificados por un decodificador de vídeo, para su uso en el proceso de decodificación de vídeo. Estos elementos sintácticos pueden proporcionarse para una PU intercodificada para permitir que el decodificador decodifique y reproduzca datos de vídeo definidos por una PU.

**[0194]** En algunos casos, los índices de referencias de la lista L0 y la lista L1 se pueden usar para señalar índices de referencia para una modalidad de predicción bidireccional (Pred\_BI), y un índice de referencia de lista combinada (LC) se usa para señalar un índice de referencia para una modalidad de predicción de unidirección (Pred\_LC). El índice de referencia LC es el índice de referencia de la lista combinada de imágenes de referencia, que incluye una combinación de imágenes de referencia de las listas L0 y L1, con las imágenes de referencia duplicadas eliminadas de acuerdo a una regla predefinida (o señalización explícita). El índice de referencia LC, por lo tanto, correlaciona con un índice de referencia para una entre la Lista L0 o L1.

**[0195]** En tales casos, el elemento sintáctico de dirección de interpredicción (indicador\_inter\_pred) solo tiene dos valores posibles (bidireccional o unidireccional de LC). Cuando el elemento sintáctico de dirección de interpredicción está binarizado, solo un binario puede necesitar ser codificado para indicar la dirección de interpredicción, ya sea como bidireccional o unidireccional. La Tabla 8, que se muestra a continuación, ilustra el elemento sintáctico de dirección de interpredicción:

**TABLA 8**

tipo_fragmento	indicador_inter_pred	Nombre de indicador_inter_pred
----------------	----------------------	-----------------------------------

P	inferido	Pred_L0
B	0	Pred_LC
	1	Pred_BI

5 [0196] En otros ejemplos, como se propone en una presentación de T. Lee y J. Park, "On Reference List Combination [Sobre la combinación de listas de referencia]", JCTVC-I0125, Ginebra, abril de 2012, la lista combinada (LC) puede eliminarse. En tales ejemplos, en cambio, utilizar un elemento sintáctico de dirección de interpredicción (indicador\_inter\_pred o inter\_pred\_idc) con tres valores posibles (bidirección, unidireccional desde L0 o unidirección desde L1). Cuando la modalidad de predicción es la modalidad de predicción unidireccional, es posible que se deba codificar un binario adicional que indique Pred\_L0 o Pred\_L1.

10 [0197] La Tabla 9 ilustra el cambio en la codificación del elemento sintáctico de la dirección de interpredicción (en relación con la Tabla 8 anterior):

TABLA 9

tipo_fragmento	inter_pred_idc	Nombre de inter_pred_idc
P	inferido	Pred_L0
B	0	Pred_L0
	1	Pred_L1
	2	Pred_BI

15 [0198] La figura 8A ilustra la estructura de codificación descrita anteriormente con respecto a la Tabla 9. Como se muestra en la figura 8A, el elemento sintáctico de dirección de interpredicción (indicador\_inter\_pred) se puede codificar con un proceso de CABAC utilizando dos binarios. El primer binario (bin (0)) indica si la modalidad de interpredicción es unidireccional (bin (0) = 0) o bidireccional (pred\_BI) (bin (0) = 1). El segundo binario se codifica condicionalmente, solo si el primer binario indica la modalidad de predicción unidireccional. El segundo binario (bin (1)) indica si la modalidad de predicción unidireccional es de la List0 (pred\_L0) (bin (1) = 0) o de la List1 (pred\_L1) (bin (1) = 1).

25 [0199] La obtención del índice del modelo de contexto para la dirección de interpredicción, que tiene solo dos valores posibles (bidireccional o unidireccional desde LC) puede determinarse en función de la profundidad de la CU, que puede tener valores en un rango de 0..3 para el primer binario (bin0), como se muestra en la siguiente ecuación:

$$ctxIdx = profundidadCU$$

30 [0200] En el ejemplo de la figura 8A, el segundo binario (bin (1)) puede codificarse con un contexto adicional, o puede codificarse reutilizando un contexto del primer binario (bin (0)). Sin embargo, la introducción de un contexto adicional puede aumentar la complejidad asociada a la codificación del elemento sintáctico de dirección de interpredicción. Además, la reutilización de uno de los contextos del primer binario puede reducir el número de contextos utilizados para la codificación de bin0, y el codificador de vídeo debe realizar una verificación adicional para esta condición.

35 [0201] La figura 8B es un diagrama de bloques que ilustra la codificación por omisión del elemento sintáctico de dirección de interpredicción, de acuerdo a aspectos de esta divulgación. Como se muestra en la figura 8B, el primer binario (bin (0)) del elemento sintáctico de dirección de interpredicción (indicador\_inter\_pred) puede codificarse por contexto con un proceso de CABAC, y el segundo binario (bin (1)) puede codificarse con una modalidad de omisión del proceso de CABAC. El primer binario (bin (0)) indica si la modalidad de interpredicción es unidireccional (bin (0) = 0) o bidireccional (pred\_BI) (bin (0) = 1). En este ejemplo, bin (0) puede codificarse utilizando uno de los cuatro contextos posibles, ctxIdx = 0..3. El segundo binario (binario (1)) puede codificarse condicionalmente solo si el primer binario indica la modalidad de predicción unidireccional. El segundo binario (bin (1)) indica si la modalidad de predicción unidireccional es de la List0 (pred\_L0) (bin (1) = 0) o de la List1 (pred\_L1) (bin (1) = 1). De acuerdo a aspectos de esta divulgación, bin (1) puede codificarse sin usar ningún contexto (por ejemplo, codificarse utilizando una modalidad de omisión del proceso de CABAC).

40 [0202] De esta manera, de acuerdo a aspectos de esta divulgación, un codificador de vídeo (por ejemplo, un codificador de vídeo 20 o un decodificador de vídeo 30), puede binarizar un valor de dirección de interpredicción y codificar al menos un binario del valor binarizado de dirección de interpredicción con una modalidad de omisión. Más específicamente, el codificador de vídeo puede seleccionar un contexto para codificar por contexto el primer binario (bin (0)) para el valor de dirección de interpredicción con un proceso de CABAC, y codificar el segundo binario (bin (1)) con una modalidad de omisión del proceso de CABAC. Alternativamente, las técnicas permiten al  
 55 codificador de vídeo codificar el primer binario (bin (0)) para el valor de la dirección de interpredicción con la

modalidad de omisión y también codificar el segundo binario (bin (1)) con la modalidad de omisión. En consecuencia, los cuatro contextos disponibles para codificar bin (0),  $ctxIdx = 0..3$ , se pueden conservar.

5 **[0203]** Las técnicas permiten la codificación del elemento sintáctico de dirección de interpredicción con tres valores posibles (bidireccional, unidireccional desde L0 o unidireccional desde L1) sin requerir ningún contexto adicional ni reutilizar el contexto (por ejemplo, el contexto de bin0). Además, las técnicas utilizan la modalidad de omisión sin ningún contexto requerido, lo que puede ser menos complejo en términos de cálculo que la codificación por contexto.

10 **[0204]** La figura 9 es un diagrama de flujo que ilustra un ejemplo de codificación por entropía de un valor de índice de referencia, de acuerdo a aspectos de esta divulgación. Aunque en general se describe como realizado por los componentes del codificador de vídeo 20 (Figuras 1 y 2) con fines explicativos, debería entenderse que otras unidades de codificación de vídeo, procesadores, unidades de procesamiento, unidades de codificación basadas en hardware, tales como codificadores/decodificadores (CÓDEC), y similares, también pueden configurarse para realizar el proceso de la figura 9.

15 **[0205]** En el ejemplo de la figura 9, el codificador de vídeo 20 incluye un elemento sintáctico recibido (180). El codificador de vídeo 20 puede binarizar el elemento sintáctico de acuerdo a cualquiera de los procesos de binarización descritos en esta divulgación. Los procesos ejemplares de binarización incluyen el unario, el truncado unario, el exponencial-Golomb o similares.

20 **[0206]** El codificador de vídeo 20 determina si el elemento sintáctico binarizado es un valor de índice de referencia (182). Nuevamente, un valor de índice de referencia generalmente identifica una imagen de referencia en una lista de imágenes de referencia con fines de interpredicción. Si el elemento sintáctico binarizado es un valor de índice de referencia (la rama "sí" de la etapa 182), el codificador de vídeo 20 puede codificar al menos un binario del valor de índice de referencia binarizado, utilizando codificación adaptativa al contexto, tal como CABAC (184). Además, el codificador de vídeo 20 puede codificar al menos otro binario del valor de índice de referencia binarizado (en los casos en los que hay binarios adicionales que codificar) utilizando codificación por omisión, que omite el motor de codificación adaptativa al contexto (186).

25 **[0207]** Como se ha descrito anteriormente con respecto a la figura 4, el codificador de vídeo 20 puede, en algunos ejemplos, codificar uno, dos o tres binarios usando codificación por contexto. Para los binarios codificados por contexto, el codificador de vídeo 20 puede seleccionar un contexto basándose en una posición relativa del binario en la cadena de binarios. Por ejemplo, el codificador de vídeo 20 puede seleccionar un contexto para un primer binario que es diferente a un contexto para un segundo binario.

30 **[0208]** En cualquier caso, el codificador de vídeo 20 puede combinar los binarios codificados por contexto y los binarios codificados por omisión para formar un flujo de bits codificado 188. En algunos ejemplos, si el elemento sintáctico que se está codificando no es un valor de índice de referencia (la rama "no" de la etapa 182), el codificador de vídeo 20 puede seleccionar una modalidad de codificación particular (por ejemplo, de omisión o adaptativa al contexto) para codificar el elemento sintáctico (190). El codificador de vídeo 20 puede codificar el elemento sintáctico utilizando la modalidad seleccionada (192) y formar un flujo de bits codificado (188).

35 **[0209]** Se debería entender también que las etapas mostradas y descritas con respecto a la figura 9 se proporcionan simplemente como un ejemplo. Es decir, las etapas del procedimiento de la figura 9 no necesariamente se deben realizar en el orden mostrado en la figura 9, y se pueden realizar menos etapas, etapas adicionales o alternativas. Por ejemplo, el codificador de vídeo 20 puede, en algunos casos, determinar si el elemento sintáctico es un índice de referencia (etapa 182) antes de binarizar el elemento sintáctico (paso 180).

40 **[0210]** La figura 10 es un diagrama de flujo que ilustra un ejemplo de decodificación por entropía de un valor de índice de referencia, de acuerdo a aspectos de esta divulgación. Aunque en general se describe como realizado por los componentes del decodificador de vídeo 30 (Figuras 1 y 3) con fines explicativos, debería entenderse que otras unidades de codificación de vídeo, procesadores, unidades de procesamiento, unidades de codificación basadas en hardware, tales como codificadores/decodificadores (CODEC), y similares, también pueden configurarse para realizar el proceso de la figura 12.

45 **[0211]** El decodificador de vídeo 30 puede analizar inicialmente un elemento sintáctico codificado de un flujo de bits codificado (200). Por ejemplo, el decodificador de vídeo 30 puede leer y segmentar un elemento sintáctico codificado de un flujo de bits codificado, de acuerdo a un proceso particular de análisis sintáctico (por ejemplo, análisis sintáctico de frentes de onda). El elemento sintáctico codificado puede incluir una pluralidad de binarios codificados, es decir, valores binarios.

50 **[0212]** El decodificador de vídeo 30 también puede determinar si parte del flujo de bits que se está decodificando actualmente es un valor de índice de referencia (202). Si el decodificador de vídeo 30 está decodificando un valor de índice de referencia (la rama "sí" de la etapa 202), el decodificador de vídeo 30 puede decodificar al menos un binario utilizando la codificación adaptativa al contexto (206). Además, el decodificador de vídeo 30 puede

55

5 decodificar al menos otro binario (en los casos en que hay binarios adicionales para codificar) utilizando la codificación por omisión (208). Como se ha indicado anteriormente con respecto a la figura 9, el decodificador de vídeo 30 puede, en algunos ejemplos, codificar uno, dos o tres binarios usando codificación por contexto. Para los binarios codificados por contexto, el decodificador de vídeo 30 puede seleccionar un contexto basándose en una posición relativa del binario en la cadena de binarios. Por ejemplo, el decodificador de vídeo 30 puede seleccionar un contexto para un primer binario que es diferente a un contexto para un segundo binario.

10 **[0213]** Después de decodificar los binarios para producir valores binarios decodificados, el decodificador de vídeo 30 puede binarizar la cadena de binarios decodificados para producir un elemento sintáctico decodificado (208). Por ejemplo, el decodificador de vídeo 30 puede correlacionar la cadena de binarios decodificados con un elemento sintáctico utilizando un proceso predeterminado. Es decir, en algunos casos, el decodificador de vídeo 30 puede recibir una indicación de que un binario particular es el binario final para un elemento sintáctico. Al completar el elemento sintáctico, entonces, el decodificador de vídeo 30 puede correlacionar la cadena de binarios con un elemento sintáctico utilizando una tabla de binarización.

15 **[0214]** En algunos ejemplos, si el elemento sintáctico que se está codificando no es un valor de índice de referencia (la rama "no" de la etapa 202), el decodificador de vídeo 30 puede seleccionar una modalidad de codificación particular (por ejemplo, por omisión o adaptativa al contexto) para codificar el elemento sintáctico (210). El decodificador de vídeo 30 puede decodificar el elemento sintáctico usando la modalidad seleccionada (212) y binarizar la cadena de binarios decodificados (208).

20 **[0215]** Se debería entender también que las etapas mostradas y descritas con respecto a la figura 10 se proporcionan simplemente como un ejemplo. Es decir, las etapas del procedimiento de la figura 10 no necesariamente se deben realizar en el orden mostrado en la figura 10, y se pueden realizar menos etapas, etapas adicionales o alternativas.

25 **[0216]** La figura 11 es un diagrama de flujo que ilustra un ejemplo de codificación por entropía de datos de predicción, de acuerdo a aspectos de esta divulgación. Aunque en general se describe como realizado por los componentes del codificador de vídeo 20 (Figuras 1 y 2) con fines explicativos, debería entenderse que otras unidades de codificación de vídeo, procesadores, unidades de procesamiento, unidades de codificación basadas en hardware, tales como codificadores/decodificadores (CÓDEC), y similares, también pueden configurarse para realizar el proceso de la figura 9.

30 **[0217]** En el ejemplo de la figura 11, el codificador de vídeo 20 puede binarizar uno o más elementos sintácticos actualmente codificados (220). Por ejemplo, el codificador de vídeo 20 puede binarizar datos de predicción que incluyen uno o más índices de referencia, vectores de movimiento, predictores de vectores de movimiento, índices de predictores de vectores de movimiento, valores de diferencia de vectores de movimiento y similares.

35 **[0218]** En cualquier caso, el codificador de vídeo 20 puede determinar si los elementos sintácticos que se codifican incluyen binarios para la codificación por contexto y binarios para la codificación por omisión (222). Es decir, el codificador de vídeo 20 puede determinar si los binarios de un elemento sintáctico están codificados utilizando una combinación de codificación adaptativa al contexto y codificación por omisión. Si hay una mezcla tanto de codificación por contexto y como de codificación por omisión (la rama "sí" de la etapa 222), el codificador de vídeo 20 puede agrupar los binarios codificados por contexto y los binarios codificados por omisión (224). Por ejemplo, el codificador de vídeo 20 puede separar los binarios codificados por contexto de los binarios codificados por omisión.

40 **[0219]** El codificador de vídeo 20 puede luego codificar los binarios codificados por contexto utilizando, por ejemplo, un proceso de codificación adaptativa al contexto (por ejemplo, tal como CABAC) (226). Además, el codificador de vídeo 20 puede codificar los binarios codificados por omisión utilizando una modalidad por omisión (226). La modalidad de omisión puede omitir el motor de codificación adaptativa al contexto y usar una probabilidad fija para codificar los binarios.

45 **[0220]** Si el elemento sintáctico que se está codificando no incluye tanto los binarios codificados por contexto como los binarios codificados por omisión (la rama "no" de la etapa 222), el codificador de vídeo 20 puede seleccionar una modalidad de codificación particular (por ejemplo, por omisión o adaptativa al contexto) para codificar el elemento sintáctico (230). El codificador de vídeo 20 puede codificar entonces el elemento sintáctico utilizando la modalidad seleccionada (234).

50 **[0221]** En algunos casos, la agrupación de binarios descrita con respecto a la figura 11 se puede realizar para más de dos elementos sintácticos. Por ejemplo, como se ha descrito anteriormente con respecto a la figura 7, todos los binarios codificados por contexto asociadas a una PU pueden agruparse, de modo tal que los binarios codificados por contexto de la PU se codifiquen juntos y los binarios codificados por omisión de la PU se codifiquen juntos. Además, la agrupación se puede realizar en un nivel de CU, de LCU o de fragmento. Es decir, en algunos ejemplos, todos los binarios codificados por contexto para una CU / una LCU / un fragmento pueden agruparse y codificarse juntos, permitiendo por ello que el codificador de vídeo 20 realice una transición única entre la

codificación por contexto y la codificación por omisión.

**[0222]** Se debe entender también que las etapas mostradas y descritas con respecto a la figura 11 se proporcionan simplemente como un ejemplo. Es decir, las etapas del procedimiento de la figura 11 no necesariamente se deben realizar en el orden mostrado en la figura 11, y se pueden realizar menos etapas, etapas adicionales o alternativas. Por ejemplo, mientras que la figura 11 muestra el codificador de vídeo 20 codificando binarios codificados por contexto antes de los binarios codificados por omisión, en otros ejemplos, el codificador de vídeo 20 puede codificar binarios codificados por omisión antes de los binarios codificados por contexto.

**[0223]** La figura 12 es un diagrama de flujo que ilustra un ejemplo de decodificación por entropía de datos de predicción, de acuerdo a aspectos de esta divulgación. Aunque en general se describe como realizado por los componentes del decodificador de vídeo 30 (Figuras 1 y 3) con fines explicativos, debería entenderse que otras unidades de codificación de vídeo, procesadores, unidades de procesamiento, unidades de codificación basadas en hardware, tales como codificadores/decodificadores (CODEC), y similares, también pueden configurarse para realizar el proceso de la figura 12.

**[0224]** En el ejemplo de la figura 12, el decodificador de vídeo 30 puede determinar el elemento sintáctico (o elementos) a decodificar (240). En un ejemplo, el decodificador de vídeo 30 puede identificar elementos sintácticos asociados a datos de predicción para su decodificación. El decodificador de vídeo 30 también puede determinar si los binarios de los uno o más elementos sintácticos incluyen binarios codificados por contexto y binarios codificados por omisión (242). Si hay una mezcla tanto de codificación por contexto como de codificación por omisión (la rama "sí" de la etapa 242), el decodificador de vídeo 30 puede decodificar los binarios codificados por contexto (244). El decodificador de vídeo 30 también puede decodificar, por separado de los binarios codificados por contexto, los binarios codificados por omisión (246). Es decir, en los casos en que los binarios codificados por contexto se agrupan por separado de los binarios codificados por omisión en el flujo de bits que se está decodificando, el decodificador de vídeo 30 puede decodificar todo el contexto codificado por separado de los binarios codificados por omisión. Después de decodificar los binarios, el decodificador de vídeo 30 puede binarizar los binarios decodificados para formar un elemento sintáctico decodificado (248). Por ejemplo, el decodificador de vídeo 30 puede correlacionar la cadena de binarios decodificados con un elemento sintáctico utilizando una tabla de binarización u otro proceso de binarización.

**[0225]** Si el elemento sintáctico que se está codificando no incluye tanto los binarios codificados por contexto como los binarios codificados por omisión (la rama "no" de la etapa 242), el decodificador de vídeo 30 puede seleccionar una modalidad de codificación particular (por ejemplo, por omisión o adaptativa al contexto) para codificar el elemento sintáctico (250). El decodificador de vídeo 30 puede luego decodificar los binarios codificados utilizando la modalidad seleccionada (252), y binarizar la cadena de binarios decodificados para formar un elemento sintáctico decodificado (248).

**[0226]** Como se ha descrito anteriormente con respecto a la figura 11, en algunos casos, la agrupación de binarios se puede realizar para más de dos elementos sintácticos. Por ejemplo, la agrupación se puede realizar en un nivel de PU, de CU, de LCU o de fragmento. Es decir, en algunos ejemplos, todos los binarios codificados por contexto para una PU / una CU / una LCU / un fragmento se pueden agrupar y codificar juntos, permitiendo así que el decodificador de vídeo 30 realice una transición única entre la codificación por contexto y la codificación por omisión.

**[0227]** Se debería entender también que las etapas mostradas y descritas con respecto a la figura 12 se proporcionan simplemente como un ejemplo. Es decir, las etapas del procedimiento de la figura 12 no necesariamente se deben realizar en el orden mostrado en la figura 12, y se pueden realizar menos etapas, etapas adicionales o alternativas. Por ejemplo, mientras que la figura 12 muestra el decodificador de vídeo 30 decodificando los binarios codificados por contexto antes de los binarios codificados por omisión, en otros ejemplos, el decodificador de vídeo 30 puede decodificar los binarios codificados por omisión antes de los binarios codificados por contexto.

**[0228]** Debería entenderse que, según el ejemplo, ciertos actos o sucesos de cualquiera de los procedimientos descritos en el presente documento pueden realizarse en una secuencia distinta, pueden añadirse, combinarse u omitirse por completo (por ejemplo, no todos los actos o sucesos descritos son necesarios para la puesta en práctica del procedimiento). Además, en determinados ejemplos, los actos o sucesos se pueden llevar a cabo simultáneamente, por ejemplo, mediante procesamiento de múltiples hilos, procesamiento de interrupciones o múltiples procesadores, en lugar de secuencialmente. Además, si bien determinados aspectos de la presente divulgación se describen como que son llevados a cabo por un único módulo o unidad con fines de claridad, se debería entender que las técnicas de la presente divulgación pueden llevarse a cabo por una combinación de unidades o módulos asociados a un codificador de vídeo.

**[0229]** El codificador de vídeo 20 puede implementar cualquiera de, o todas, las técnicas de esta divulgación para codificar un índice de referencia y otros elementos sintácticos en un proceso de codificación de vídeo. Del mismo modo, el decodificador de vídeo 30 puede implementar cualquiera de, o todas, estas técnicas para la

codificación de índices de referencia y otros elementos sintácticos en un proceso de codificación de vídeo. Un codificador de vídeo, como se describe en esta divulgación, puede referirse a un codificador de vídeo o a un decodificador de vídeo. De manera similar, una unidad de codificación de vídeo puede referirse a un codificador de vídeo o a un decodificador de vídeo. Asimismo, la codificación de vídeo puede referirse a la codificación de vídeo o a la decodificación de vídeo.

**[0230]** En uno o más ejemplos, las funciones descritas en esta divulgación y atribuidas al codificador de vídeo 20, al decodificador de vídeo 30 o a cualquier otra unidad de procesamiento pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en, o transmitir por, un medio legible por ordenador, como una o más instrucciones o código, y ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que correspondan a un medio tangible tal como medios de almacenamiento de datos, o medios de comunicación que incluyan cualquier medio que facilite la transferencia de un programa informático desde un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder en general a (1) medios de almacenamiento tangibles legibles por ordenador que sean no transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser medios disponibles cualesquiera a los que se pueda acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

**[0231]** A modo de ejemplo, y no de limitación, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda usarse para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Además, cualquier conexión recibe adecuadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde una sede de la Red, un servidor u otro origen remoto usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. El término disco, como se usa en el presente documento, incluye el disco compacto (CD), el disco láser, el disco óptico, el disco versátil digital (DVD), el disco flexible y el disco Blu-ray, donde algunos discos normalmente emiten datos magnéticamente, mientras que otros discos emiten datos ópticamente con láseres. Las combinaciones de lo anterior también deberían incluirse dentro del alcance de los medios legibles por ordenador.

**[0232]** Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), formaciones de compuertas programables in situ (FPGA) u otros circuitos lógicos, integrados o discretos, equivalentes. El codificador de vídeo 20 o el decodificador de vídeo 30 pueden incluir cualquiera entre una diversidad de uno o más procesadores configurados para realizar las funciones descritas en esta divulgación. En consecuencia, el término "procesador", como se usa en el presente documento, se puede referir a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar dentro de módulos de hardware y/o software dedicados, configurados para la codificación y la decodificación, o incorporados en un códec combinado. Además, las técnicas se podrían implementar totalmente en uno o más circuitos o elementos lógicos.

**[0233]** Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluidos un equipo manual inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). Diversos componentes, módulos o unidades se describen en esta divulgación para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En cambio, como se ha descrito anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec, o ser proporcionadas por un grupo de unidades de hardware interoperativas, incluyendo uno o más procesadores, como se ha descrito anteriormente, conjuntamente con software y/o firmware adecuados.

**[0234]** Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

## REIVINDICACIONES

1. Un procedimiento para codificar un elemento sintáctico de índice de referencia en un proceso de codificación de vídeo, el procedimiento que comprende:
- 5 binarizar (180) un valor de índice de referencia;
- codificar (184) al menos un binario del valor del índice de referencia binarizado con una modalidad de codificación por contexto de un proceso de codificación aritmética binaria adaptativa al contexto, CABAC; y
- 10 codificar (186), cuando el valor del índice de referencia binarizado comprende más binarios que el al menos un binario codificado con la modalidad codificada por contexto, al menos otro binario del valor del índice de referencia binarizado con una modalidad de codificación por omisión del proceso de CABAC, en donde codificar al menos un binario del valor del índice de referencia binarizado con la modalidad de codificación por contexto comprende:
- 15 codificar un primer binario, bin0, del valor del índice de referencia binarizado con un primer contexto, ctx0,
- 20 codificar un segundo binario, bin1, del índice de referencia binarizado con un segundo contexto, ctx1, en donde el primer contexto es diferente al segundo contexto, y
- en donde la codificación de al menos otro binario del valor del índice de referencia binarizado con la modalidad de codificación por omisión comprende:
- 25 codificar un tercer binario, bin2, y todos los binarios restantes después del tercer binario, bin2, con la modalidad de codificación por omisión.
- 30 2. El procedimiento de la reivindicación 1, en el que la binarización del valor del índice de referencia comprende la binarización del valor del índice de referencia utilizando un código combinado de Golomb unario y exponencial truncado.
- 35 3. El procedimiento de la reivindicación 1, en el que la codificación del al menos otro binario del valor de índice de referencia binarizado con la modalidad de codificación por omisión comprende la codificación del al menos otro binario con al menos uno entre un proceso unario, uno truncado unario, uno de Golomb, uno exponencial de Golomb o uno de Golomb-Rice y, preferiblemente, que comprende además:
- 40 binarizar componentes de un primer valor de diferencia de vectores de movimiento, asociado al primer índice de referencia, y componentes de un segundo valor de diferencia de vectores de movimiento, asociado a un segundo índice de referencia;
- 45 codificar una primera parte de los componentes del primer valor de diferencia de vectores de movimiento y una primera parte de los componentes del segundo valor de diferencia de vectores de movimiento con la modalidad de codificación por contexto; y
- 50 codificar una segunda parte de los componentes del primer valor de diferencia de vectores de movimiento y una segunda parte de los componentes del segundo valor de diferencia de vectores de movimiento con la modalidad de codificación por omisión y, preferiblemente, que comprende además:
- 55 agrupar la primera parte de los componentes del primer valor de diferencia de vectores de movimiento y la primera parte de los componentes del segundo valor de diferencia de vectores de movimiento en un primer grupo, para codificar con la modalidad de codificación por contexto, y
- agrupar la segunda parte de los componentes del primer valor de diferencia de vectores de movimiento y la segunda parte de los componentes del segundo valor de diferencia de vectores de movimiento en un segundo grupo, para codificar con la modalidad de codificación por omisión.
- 60 4. El procedimiento de la reivindicación 1,
- en el que la codificación de al menos un binario del valor del índice de referencia binarizado con la modalidad de codificación por contexto comprende:
- 65 codificar el tercer binario, bin2, del índice de referencia binarizado con un tercer contexto, ctx2, en donde codificar al menos otro binario del valor del índice de referencia binarizado con la modalidad de codificación por omisión comprende:

codificar todos los binarios restantes después del tercer binario, bin2, con la modalidad de codificación por omisión y, preferiblemente, en donde la codificación de al menos un binario del valor del índice de referencia binarizado con la modalidad de codificación por contexto comprende:

5 codificar un primer binario, bin0, del valor del índice de referencia binarizado con un primer contexto, ctx0, y

10 en donde la codificación de al menos otro binario del valor del índice de referencia binarizado con la modalidad de codificación por omisión comprende:

codificar todos los binarios restantes después del primer binario, bin0, con la modalidad de codificación por omisión.

15 **5.** El procedimiento de la reivindicación 1,

en el que binarizar el valor del índice de referencia comprende la codificación unaria del valor del índice de referencia y, preferiblemente, en el que la codificación unaria del valor del índice de referencia comprende la codificación unaria truncada del valor del índice de referencia.

20 **6.** El procedimiento de la reivindicación 1, en el que

la binarización del valor del índice de referencia comprende la codificación unaria y la codificación exponencial de Golomb del valor del índice de referencia;

25 en el que la codificación del al menos un binario del valor del índice de referencia binarizado con la modalidad de codificación por contexto comprende:

30 codificar al menos un binario de una parte codificada unaria del valor del índice de referencia con la modalidad de codificación por contexto; y

en el que la codificación del al menos otro binario del valor de índice de referencia binarizado con la modalidad de codificación por omisión comprende:

35 codificar al menos otro binario de la parte codificada unaria del valor de índice de referencia y una parte codificada por Golomb exponencial del valor de índice de referencia con la modalidad de codificación por omisión y, preferiblemente, comprende además truncar la parte codificada por Golomb exponencial del valor de índice de referencia antes de codificar la parte codificada por Golomb exponencial del valor del índice de referencia y, preferiblemente, en donde binarizar el valor del índice de referencia comprende codificación unaria y codificación de longitud fija del valor del índice de referencia;

45 en el que la codificación del al menos un binario del valor de índice de referencia binarizado con la modalidad de codificación por contexto comprende la codificación de al menos un binario de una parte codificada unaria del valor del índice de referencia con la modalidad de codificación por contexto; y

50 en el que la codificación del al menos otro binario del valor del índice de referencia binarizado con la modalidad de codificación por omisión comprende la codificación de al menos otro binario de la parte codificada unaria del valor del índice de referencia y una parte codificada de longitud fija del valor del índice de referencia con la modalidad de codificación por omisión y, preferiblemente, comprende además truncar la parte codificada de longitud fija del valor del índice de referencia antes de codificar la parte codificada de longitud fija del valor del índice de referencia y, preferiblemente, en donde la codificación con la modalidad de codificación por contexto comprende seleccionar uno o más modelos de probabilidad para codificar el al menos un binario del índice de referencia binarizado y la codificación del al menos un binario del índice de referencia binarizado utilizando los uno o más modelos de probabilidad seleccionados; y

55 en donde la codificación con la modalidad de codificación por omisión comprende determinar una probabilidad fija y codificar al menos otro binario del valor del índice de referencia binarizado, utilizando la probabilidad fija.

60 **7.** Un aparato para codificar un elemento sintáctico de índice de referencia en un proceso de codificación de vídeo, el aparato que comprende:

medios para binarizar un valor de índice de referencia;

65 medios para codificar al menos un binario del valor de índice de referencia binarizado con una modalidad de codificación por contexto de un proceso de codificación aritmética binaria adaptativa al contexto,

CABAC; y

medios para codificar, cuando el valor del índice de referencia binarizado comprende más binarios que el al menos un binario codificado con la modalidad de codificación por contexto, al menos otro binario del valor del índice de referencia binarizado con una modalidad de codificación por omisión del proceso de CABAC en el que codificar al menos un binario del valor del índice de referencia binarizado con la modalidad de codificación por contexto comprende:

codificar un primer binario, bin0, del valor del índice de referencia binarizado con un primer contexto, ctx0,

codificar un segundo binario, bin1, del índice de referencia binarizado con un segundo contexto, ctx1, en donde el primer contexto es diferente al segundo contexto, y

en donde la codificación de al menos otro binario del valor del índice de referencia binarizado con la modalidad de codificación por omisión comprende:

codificar un tercer binario, bin2, y todos los binarios restantes después del tercer binario, bin2, con la modalidad de codificación por omisión.

8. El aparato de la reivindicación 7, que comprende además medios para llevar a cabo el procedimiento de una cualquiera de las reivindicaciones 2 a 6.

9. Un procedimiento para decodificar un elemento sintáctico de índice de referencia en un proceso de decodificación de vídeo, el procedimiento que comprende:

decodificar (204) al menos un binario de un valor de índice de referencia con una modalidad de codificación por contexto de un proceso de codificación aritmética binaria adaptativa al contexto, CABAC;

decodificar (206), cuando el valor del índice de referencia comprende más binarios que el al menos un binario codificado con la modalidad codificada por contexto, al menos otro binario del valor del índice de referencia con una modalidad de codificación por omisión del proceso de CABAC, en donde decodificar al menos un binario del valor del índice de referencia binarizado con la modalidad de codificación por contexto comprende:

decodificar un primer binario, bin0, del valor del índice de referencia binarizado con un primer contexto, ctx0,

decodificar un segundo binario, bin1, del índice de referencia binarizado con un segundo contexto, ctx1, en donde el primer contexto es diferente al segundo contexto, y

en el que decodificar al menos otro binario del valor del índice de referencia binarizado con la modalidad de codificación por omisión comprende:

decodificar un tercer binario, bin2, y todos los binarios restantes después del tercer binario, bin2, con la modalidad de codificación por omisión; y

binarizar el valor del índice de referencia.

10. El procedimiento de la reivindicación 9, en el que binarizar el valor del índice de referencia comprende binarizar el valor del índice de referencia usando un código combinado de Golomb unario y exponencial truncado y, preferiblemente, en el que decodificar el al menos otro binario del valor del índice de referencia con la modalidad de codificación por omisión comprende decodificar el al menos otro binario con al menos uno de los procesos de codificación unario, truncado unario, de Golomb, Golomb exponencial o Golomb-Rice y, preferiblemente, comprende además:

decodificar una primera parte de componentes de un primer valor de diferencia de vectores de movimiento y una primera parte de componentes de un segundo valor de diferencia de vectores de movimiento con la modalidad de codificación por contexto;

decodificar una segunda parte de los componentes del primer valor de diferencia de vectores de movimiento y una segunda parte de los componentes del segundo valor de diferencia de vectores de movimiento con la modalidad de codificación por omisión; y

binarizar los componentes del primer valor de diferencia de vectores de movimiento, asociado al primer

índice de referencia, y los componentes del segundo valor de diferencia de vectores de movimiento, asociado a un segundo índice de referencia y, además, preferiblemente, comprende además:

5 agrupar la primera parte de los componentes del primer valor de diferencia de vectores de movimiento y la primera parte de los componentes del segundo valor de diferencia de vectores de movimiento en un primer grupo, para codificar con la modalidad de codificación por contexto, y

10 agrupar la segunda parte de los componentes del primer valor de diferencia de vectores de movimiento y la segunda parte de los componentes del segundo valor de diferencia de vectores de movimiento en un segundo grupo, para codificar con la modalidad de codificación por omisión.

**11.** El procedimiento de la reivindicación 9,

15 en el que la decodificación de al menos un binario del valor del índice de referencia con la modalidad de codificación por contexto comprende:

20 decodificar el tercer binario, bin2, del índice de referencia con un tercer contexto, ctx2, y en donde decodificar al menos otro binario del valor del índice de referencia con la modalidad de codificación por omisión comprende:

25 decodificar todos los binarios restantes después del tercer binario, bin2, con el modo de codificación por omisión y, preferiblemente, en donde decodificar al menos un binario del valor de índice de referencia con la modalidad de codificación por contexto comprende:

30 decodificar un primer binario, bin0, del valor del índice de referencia con un primer contexto, ctx0, y

en donde decodificar al menos otro binario del valor de índice de referencia con la modalidad de codificación por omisión comprende:

35 decodificar todos los binarios restantes después del primer binario, bin0, con la modalidad de codificación por omisión y, además, preferiblemente, en donde binarizar el valor del índice de referencia comprende la codificación unaria del valor del índice de referencia y, preferiblemente, en donde la codificación unaria del valor del índice de referencia comprende la codificación unaria truncada del valor del índice de referencia.

**12.** El procedimiento de la reivindicación 9,

40 en el que la decodificación del al menos un binario del valor de índice de referencia con la modalidad de codificación por contexto comprende:

45 decodificar al menos un binario de una parte codificada unaria del valor de índice de referencia con la modalidad de codificación por contexto;

en el que la decodificación del al menos otro binario del valor de índice de referencia con la modalidad de codificación por omisión comprende:

50 decodificar al menos otro binario de la parte codificada unaria del valor de índice de referencia y una parte codificada por Golomb exponencial del valor del índice de referencia con la modalidad de codificación por omisión; y

en donde la binarización del valor del índice de referencia comprende codificación unaria y codificación de Golomb exponencial del valor del índice de referencia y, preferiblemente, además comprende trincar la parte codificada según Golomb exponencial del valor del índice de referencia antes de decodificar la parte codificada según Golomb exponencial del valor del índice de referencia.

**13.** El procedimiento de la reivindicación 9,

55 en el que la decodificación del al menos un binario del valor de índice de referencia con la modalidad de codificación por contexto comprende la decodificación de al menos un binario de una parte codificada unaria del valor del índice de referencia con la modalidad de codificación por contexto;

60 en el que la decodificación del al menos otro binario del valor de índice de referencia con la modalidad de codificación por omisión comprende la decodificación de al menos otro binario del valor de índice de referencia de la parte codificada unaria y una parte codificada de longitud fija del valor de índice de referencia con la modalidad de codificación por omisión; y

65 en donde binarizar el valor del índice de referencia comprende codificación unaria y codificación de longitud

fija del valor del índice de referencia y, preferiblemente, comprende además truncar la parte codificada de longitud fija del valor del índice de referencia antes de codificar la parte codificada de longitud fija del valor del índice de referencia y, preferiblemente, en donde decodificar con la modalidad de codificación por contexto comprende seleccionar uno o más modelos de probabilidad para codificar el al menos un binario del índice de referencia binarizado y decodificar el al menos un binario del índice de referencia binarizado utilizando uno o más modelos de probabilidad seleccionados; y

5

en donde la decodificación con la modalidad de codificación por omisión comprende la determinación de una probabilidad fija y la decodificación del al menos otro binario del valor del índice de referencia binarizado utilizando la probabilidad fija.

10

**14.** Un aparato para decodificar un elemento sintáctico de índice de referencia en un proceso de decodificación de vídeo, comprendiendo el aparato medios para realizar el procedimiento de una cualquiera de las reivindicaciones 9 a 13.

15

**15.** Un medio no transitorio legible por ordenador que almacena en el mismo instrucciones que, cuando se ejecutan, causan que uno o más procesadores ejecuten el procedimiento de una cualquiera de las reivindicaciones 1 a 6 o 9 a 13.

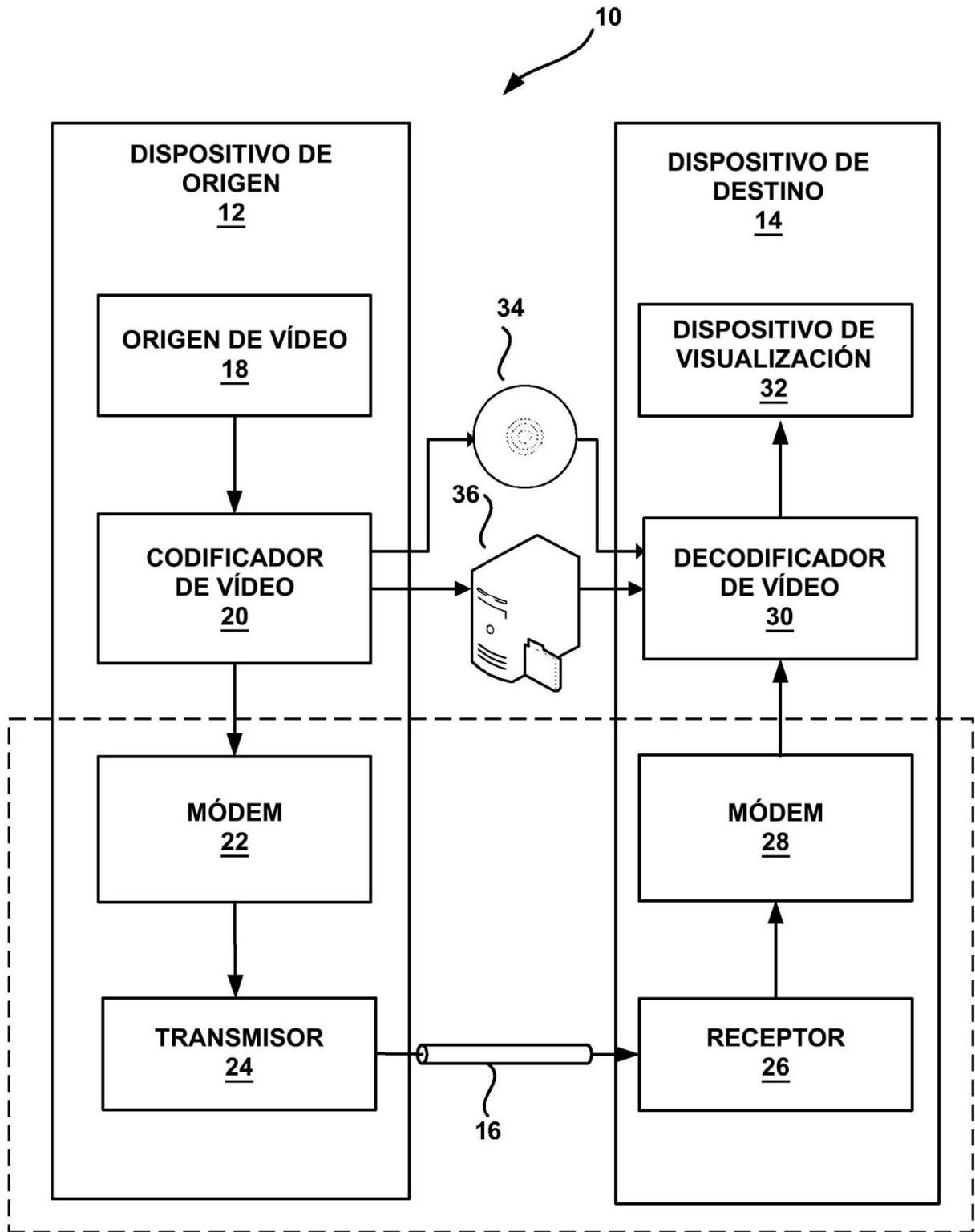


FIG. 1

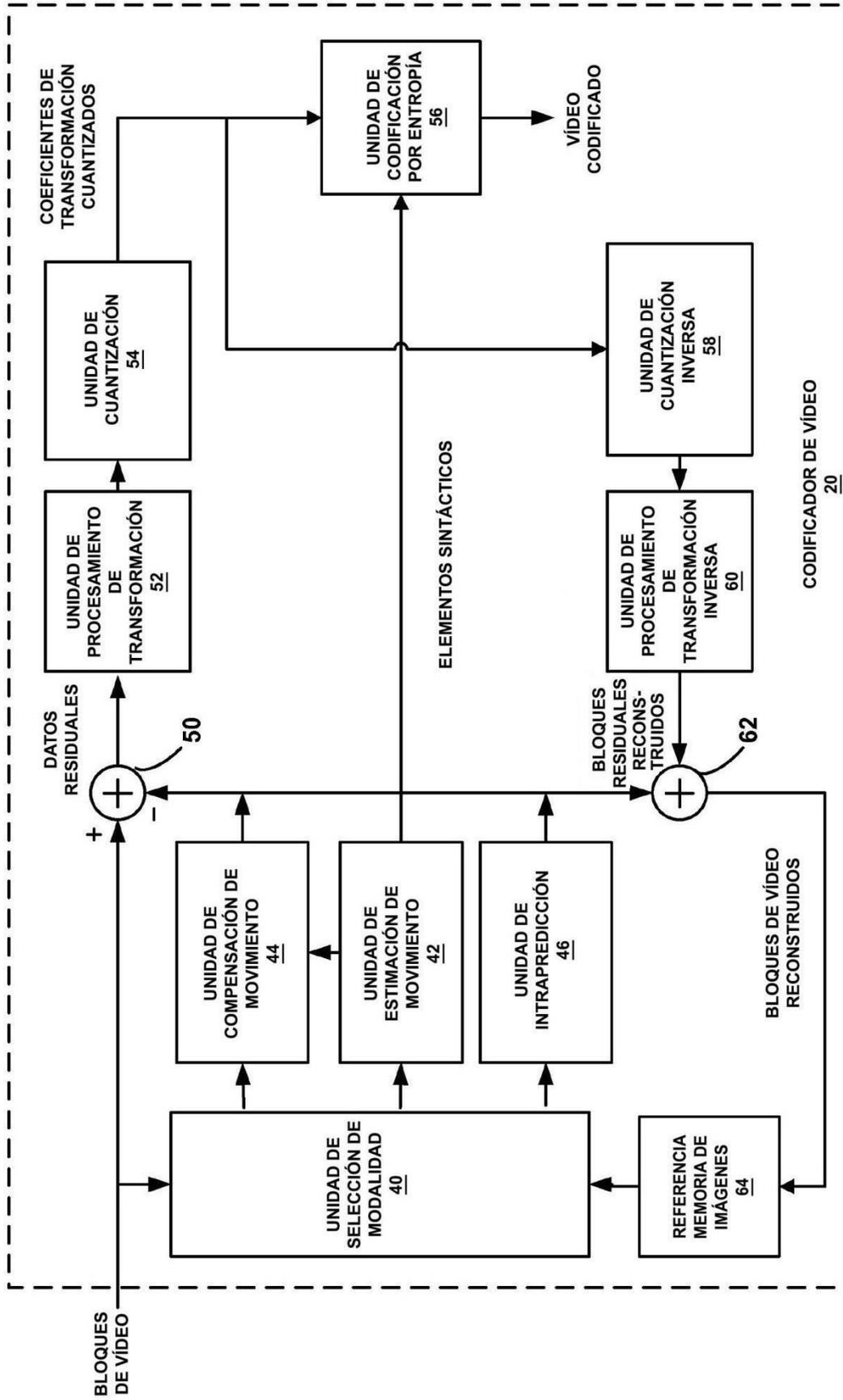


FIG. 2

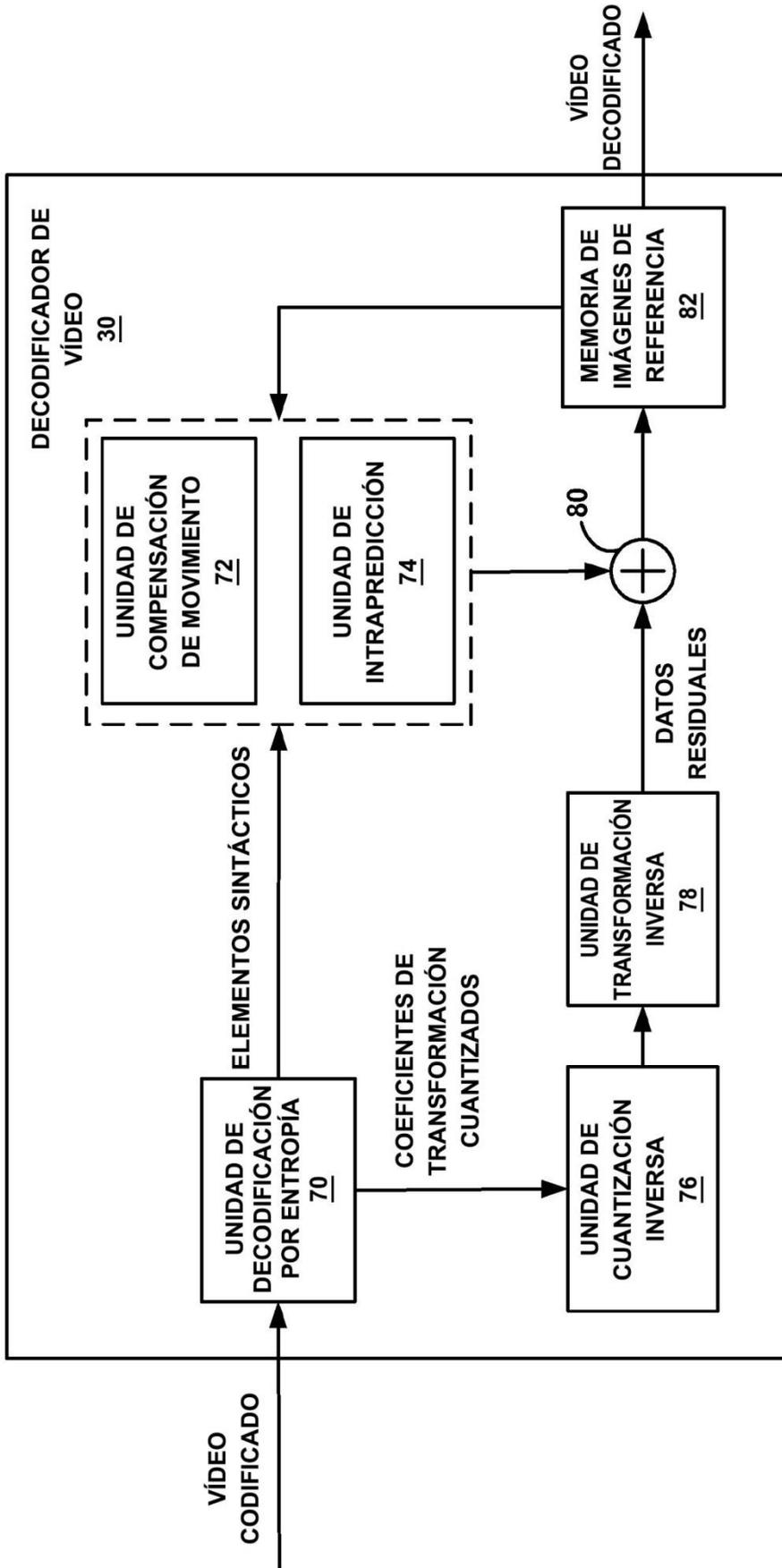


FIG. 3

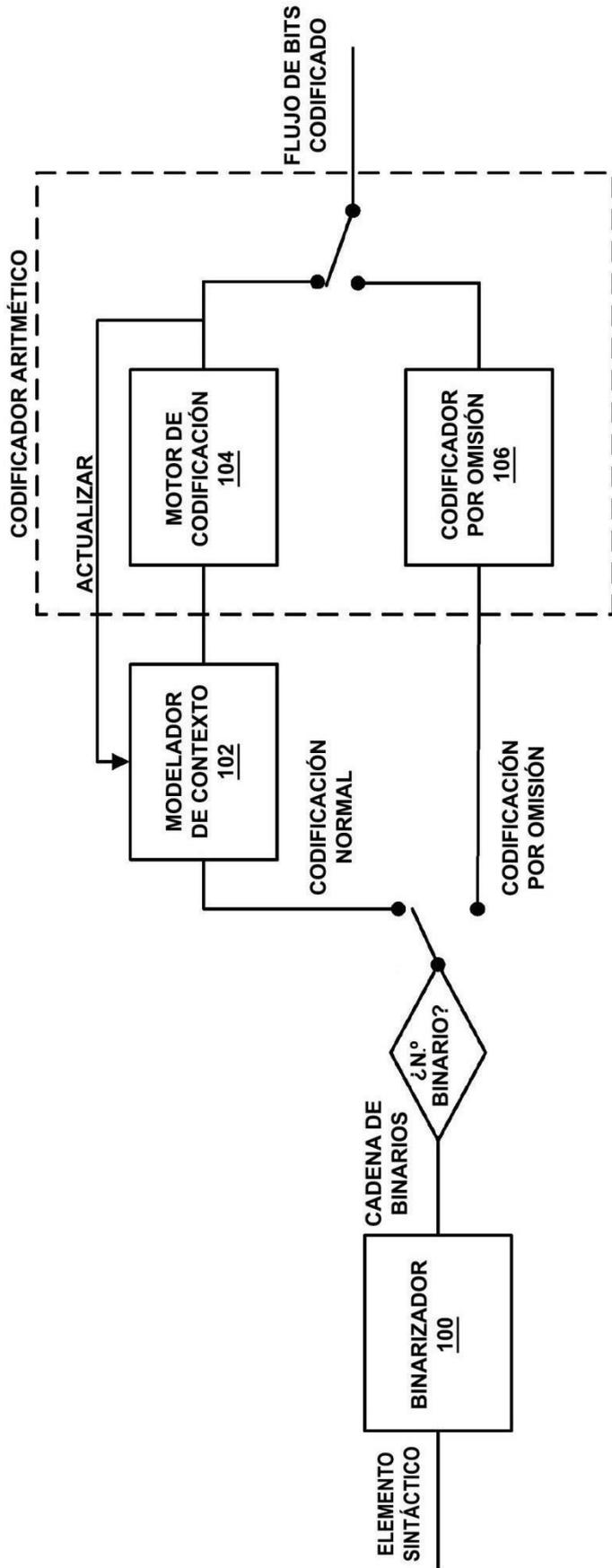


FIG. 4

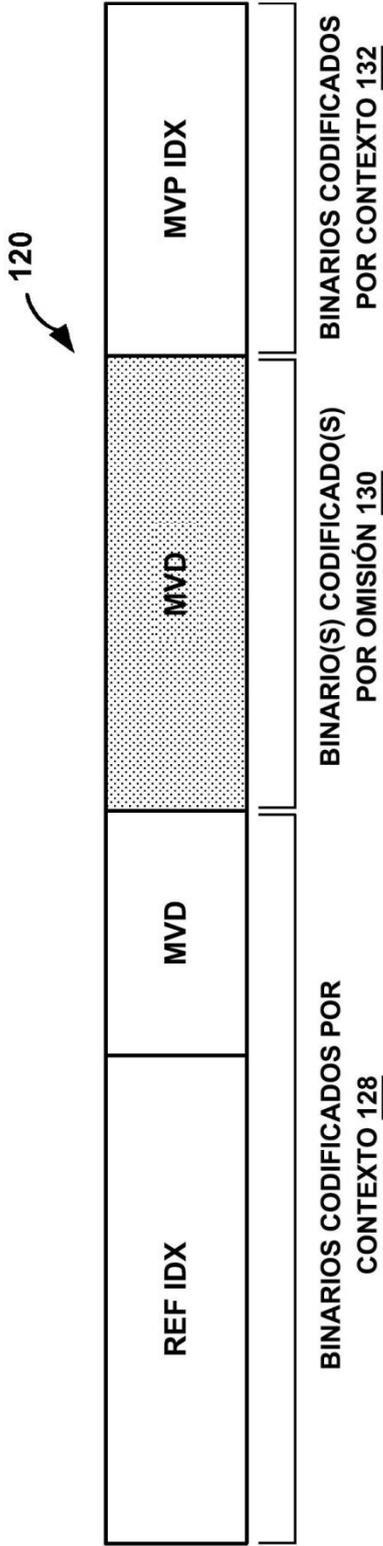


FIG. 5A

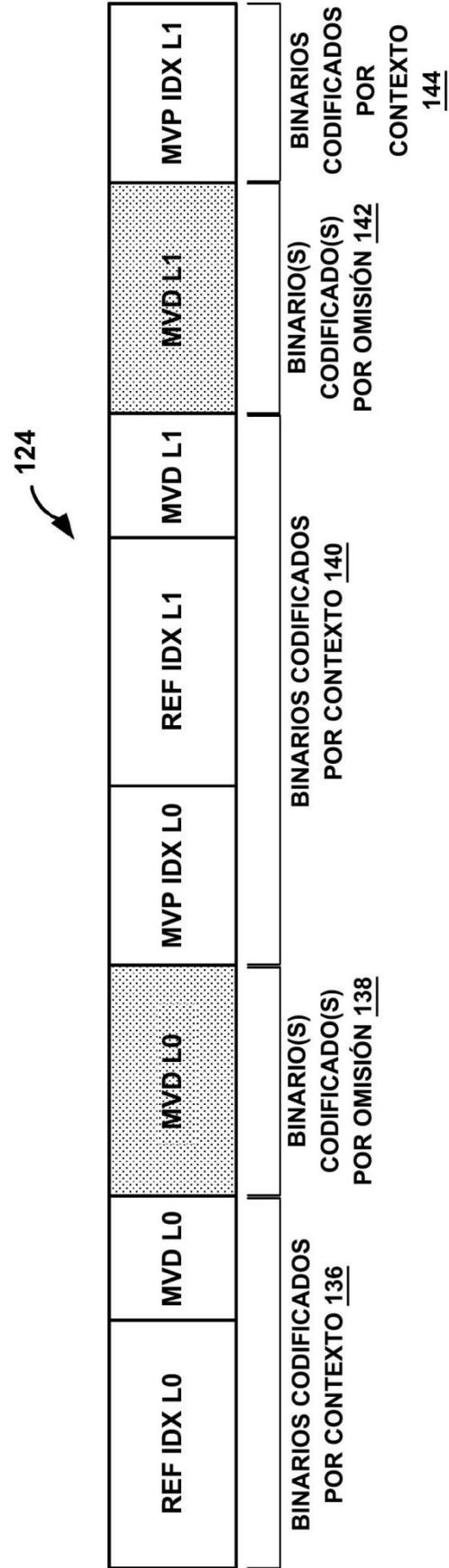


FIG. 5B

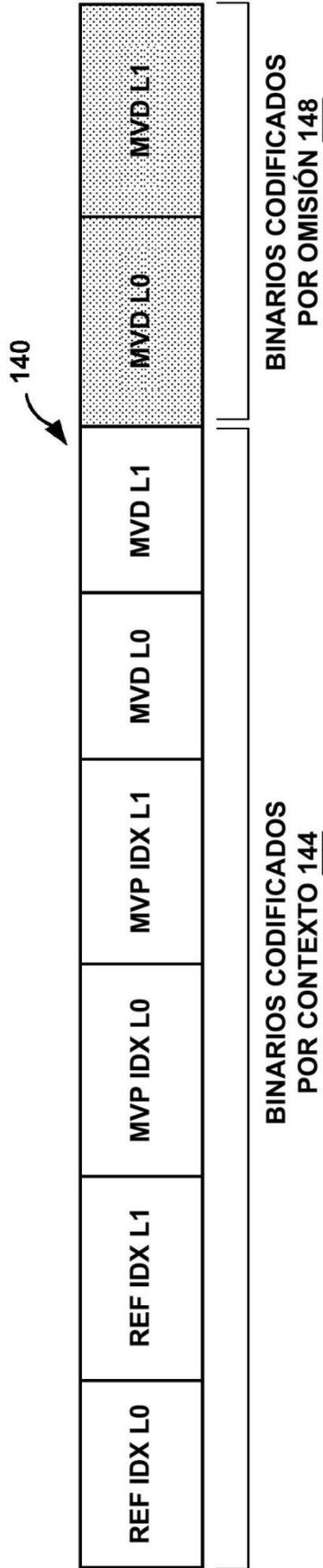


FIG. 6

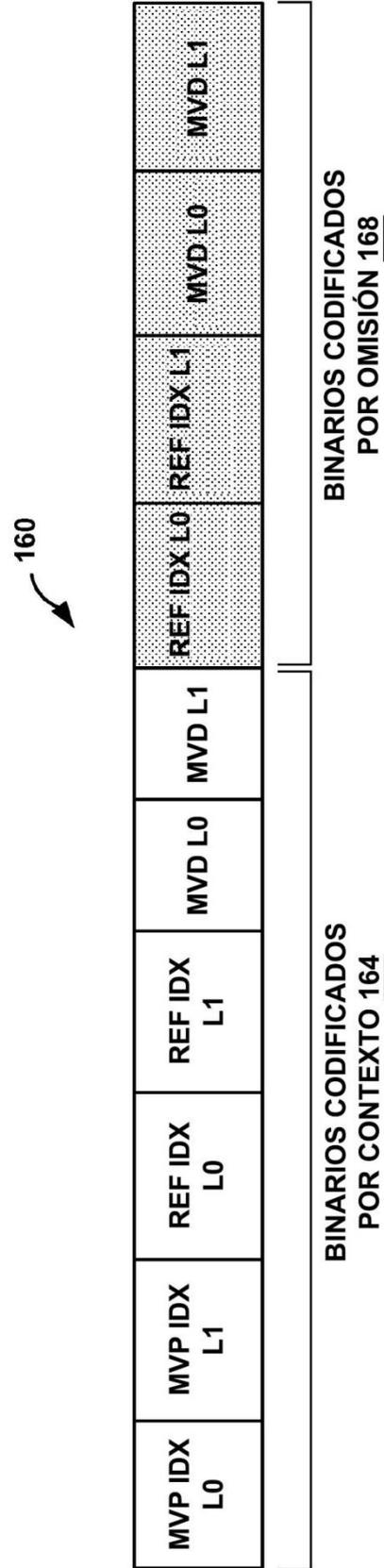
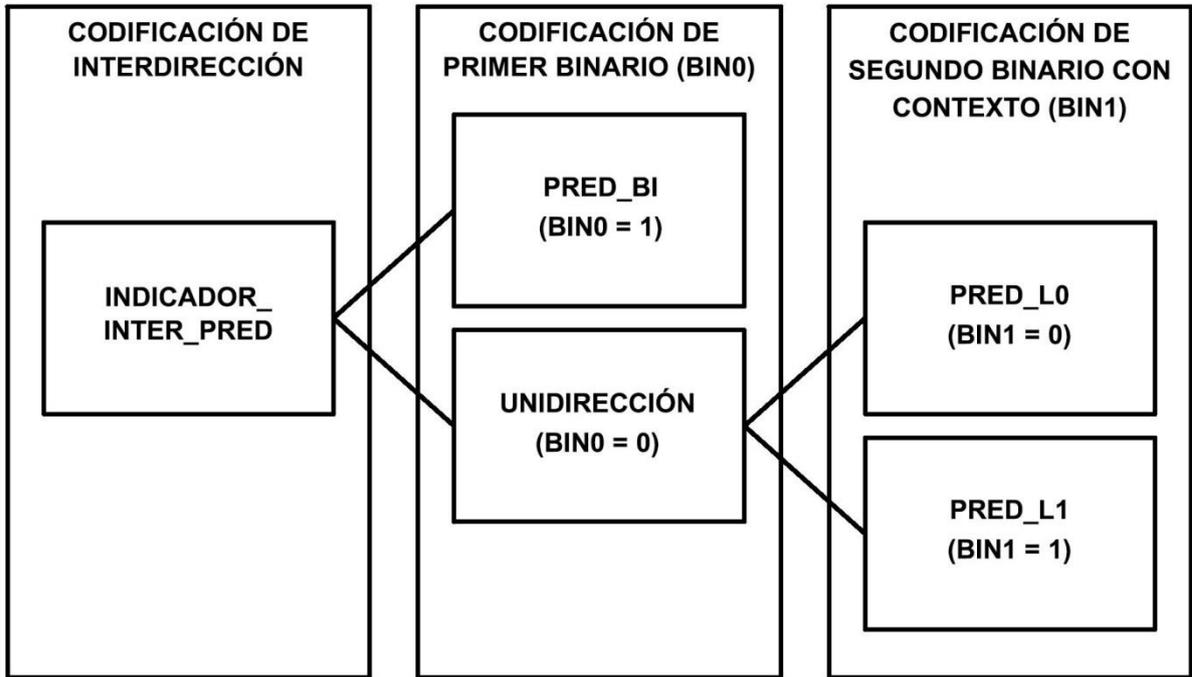
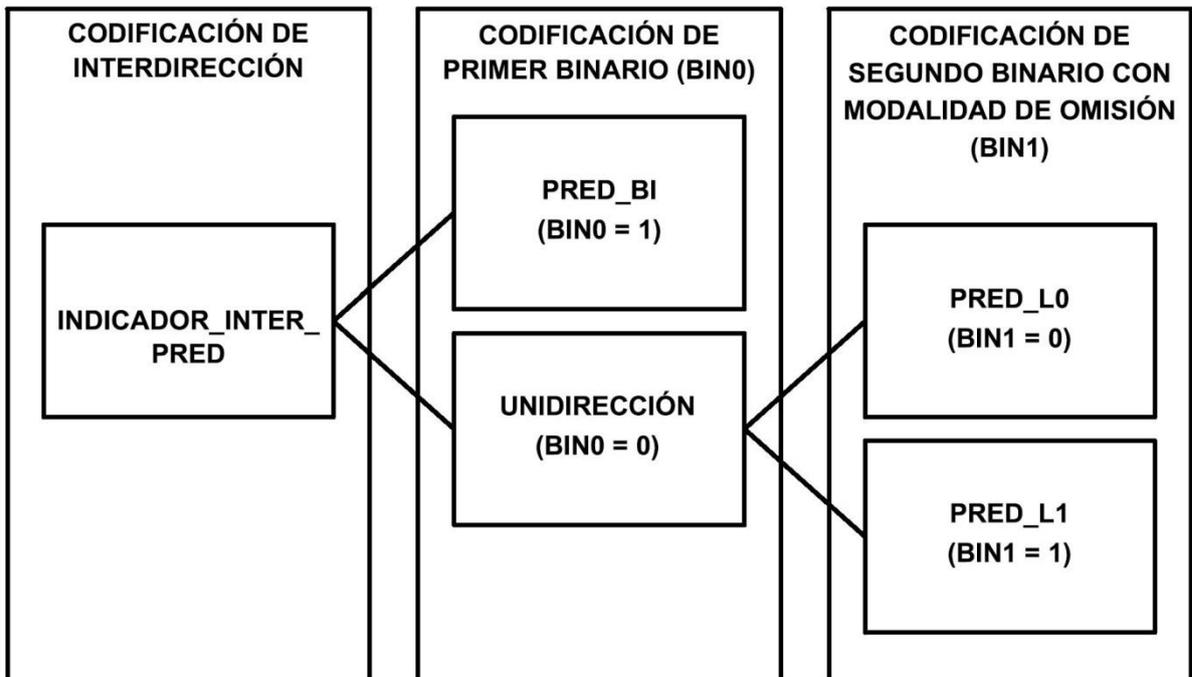


FIG. 7



**FIG. 8A**



**FIG. 8B**

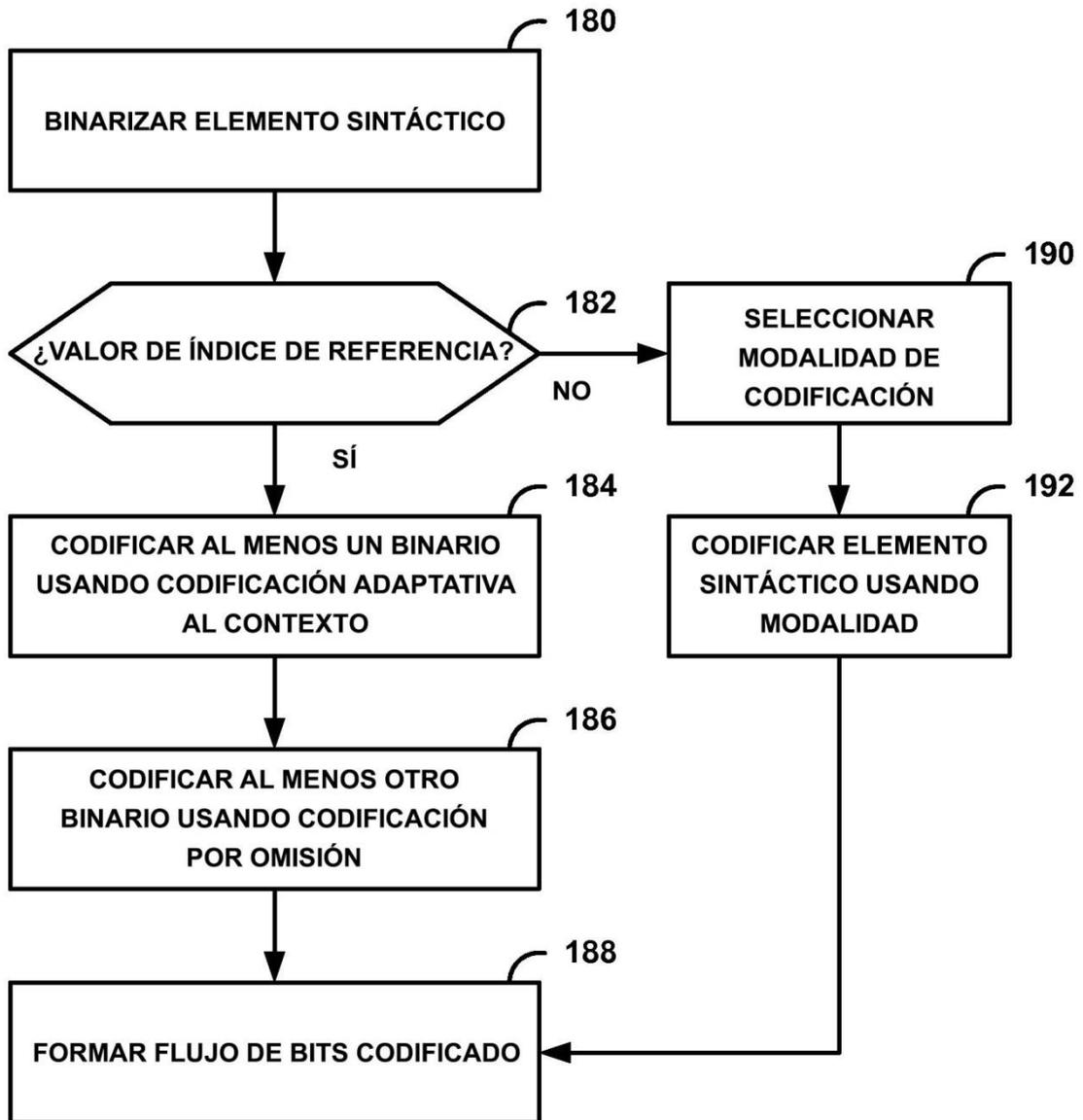


FIG. 9

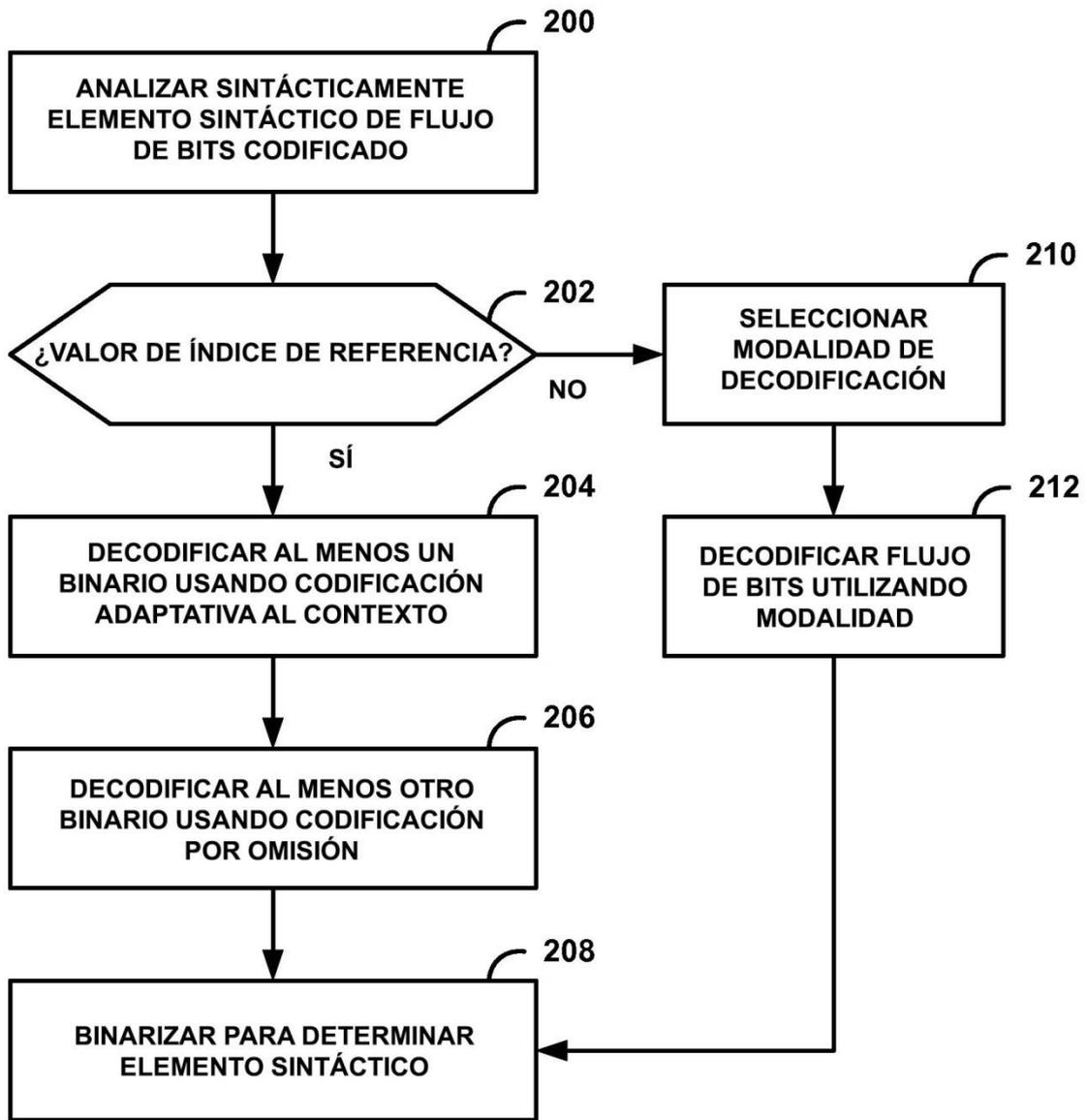


FIG. 10

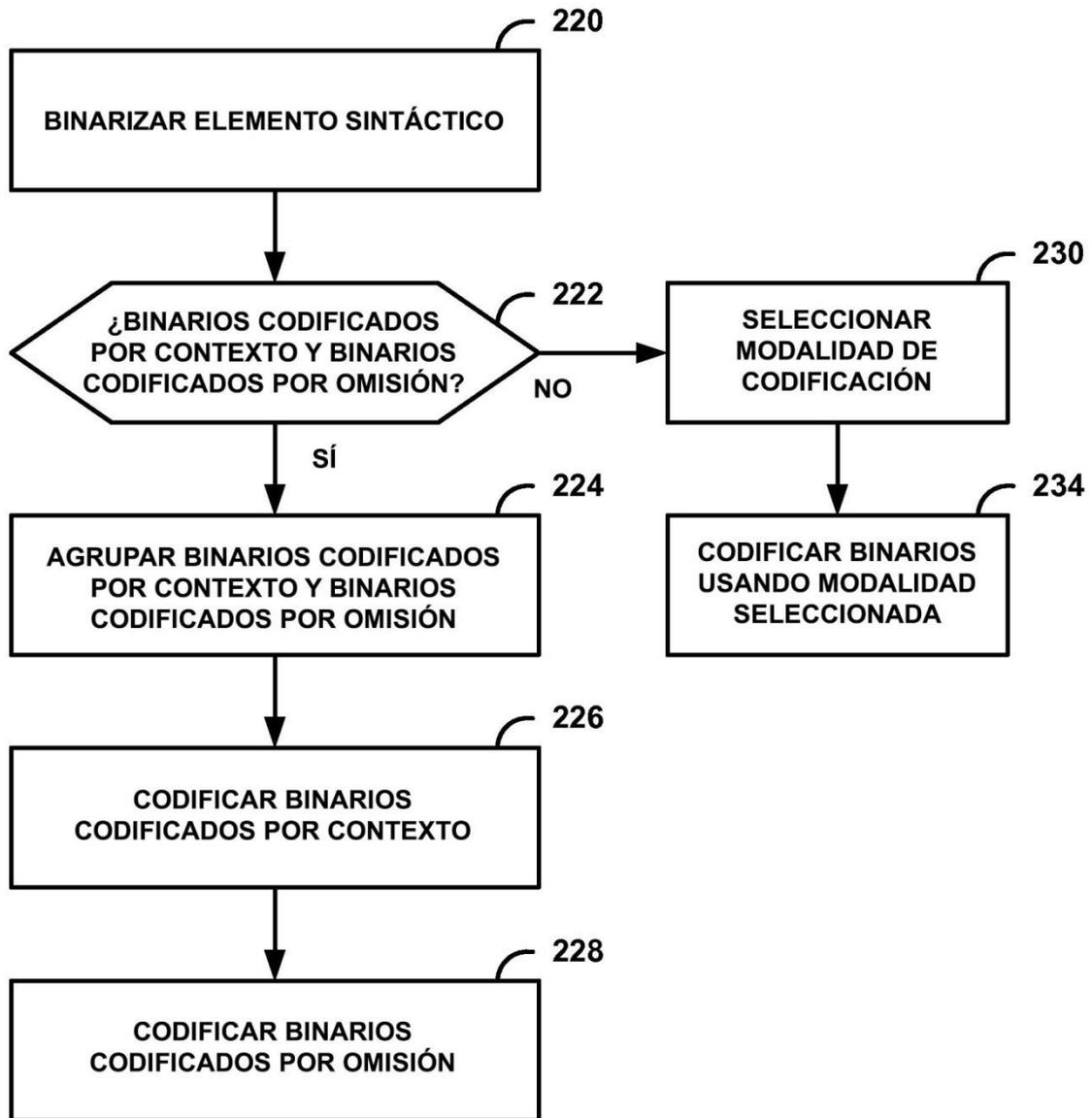


FIG. 11

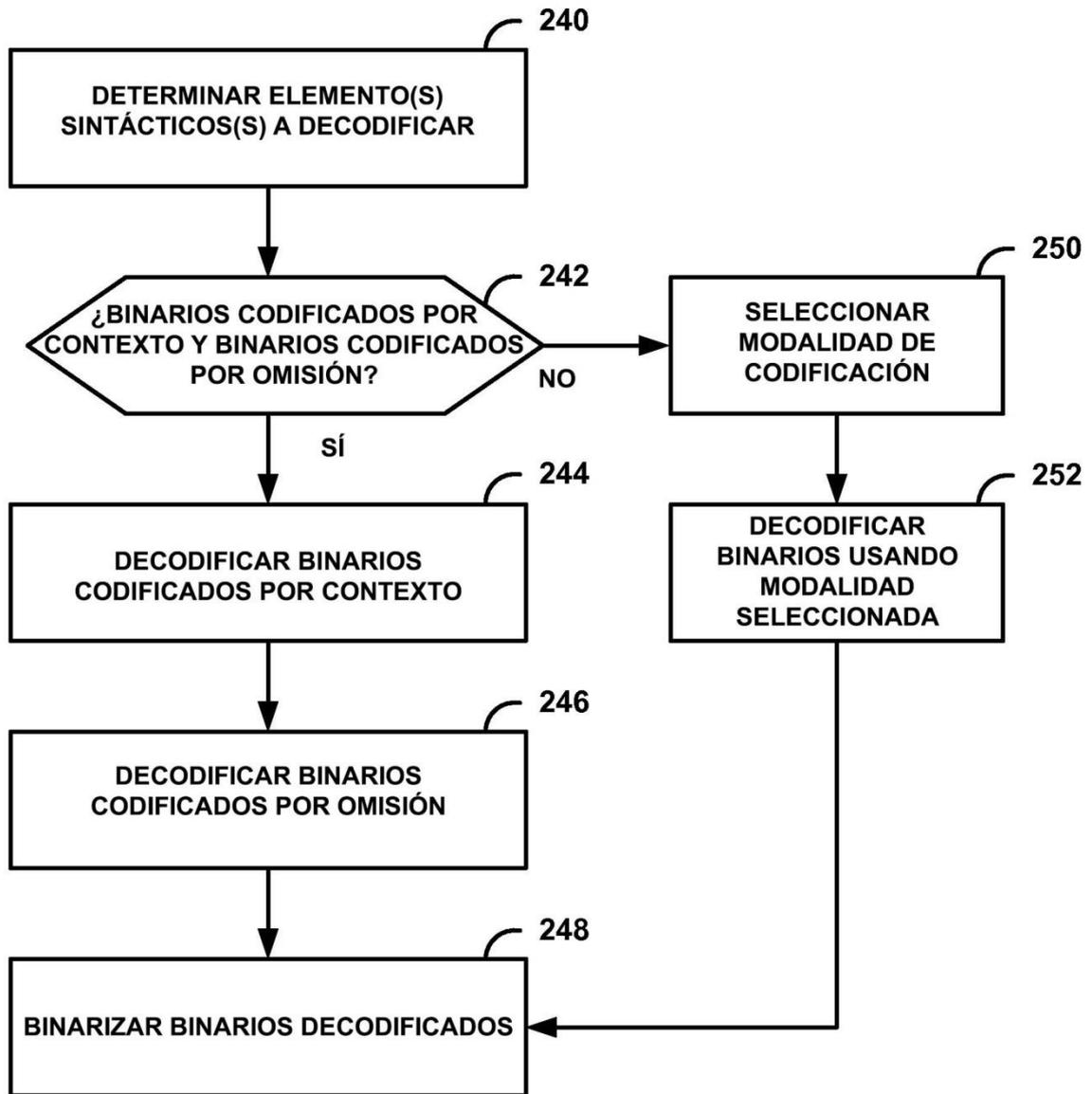


FIG. 12