

COMPARACIÓN BILATERAL ADAPTATIVA PARA EL REFINAMIENTO DEL VECTOR DE MOVIMIENTO DEL LADO DEL DECODIFICADOR PARA CODIFICACIÓN DE VIDEO

SOLICITUDES RELACIONADAS

[0001] Esta solicitud reivindica prioridad a la solicitud de patente estadounidense N.º 17/929,891, presentada el 6 de septiembre de 2022, y la solicitud provisional estadounidense N.º 63/245,577, presentada el 17 de septiembre de 2021, cuyos contenidos se incorporan en la presente por referencia en su totalidad. La solicitud de patente estadounidense N.º 17/929,891, presentada el 6 de septiembre de 2022, reivindica el beneficio de la solicitud provisional estadounidense N.º 63/245,577, presentada el 17 de septiembre de 2021.

CAMPO TÉCNICO

[0002] Esta divulgación se refiere a la codificación de video y la decodificación de video.

ANTECEDENTES

[0003] Se pueden incorporar capacidades de video digitales en un amplio rango de dispositivos, incluso televisores digitales, sistemas de transmisión directa digital, sistemas de transmisión inalámbrica, asistentes digitales personales (PDA), computadoras portátiles o de escritorio, computadora tipo *tablet*, lectores de libros electrónicos, cámaras digitales, dispositivos de registro digital, reproductores de medios digitales, dispositivos de juegos de video, consolas de juegos de video, teléfonos celulares o de radio satélite, los denominados "teléfonos inteligentes", dispositivos de teleconferencia de video, dispositivos de *streaming* de video y similares. Los dispositivos de video digital implementan técnicas de codificación de video, tales como aquellas que se describen en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, parte 10, codificación de video avanzada (AVC), ITU-T H.265/codificación de video de alta eficiencia (HEVC), ITU-T H.266/Codificación de video versátil (VVC) y extensiones de tales estándares, así como códecs/formatos de video propietarios como *AOMedia Video 1* (AV1) que fue desarrollado por *Alliance for Open Media*. Los dispositivos de video pueden transmitir, recibir, codificar, decodificar y/o almacenar información de video digital de manera más eficaz implementando las técnicas de codificación de video.

[0004] Las técnicas de codificación de video incluyen la predicción espacial

(intraimagen) y/o la predicción temporal (interimagen) para reducir o eliminar la redundancia inherente en secuencias de video. Para la codificación de video basada en bloques, un plano de video (por ejemplo, una imagen de video o un segmento de una imagen de video) se pueden dividir en bloques de video, que también se pueden denominar unidades de árboles de codificación (CTU), unidades de codificación (CU) y/o nodos de codificación. Los bloques de video en un segmento intracodificado (I) de una imagen se codifican mediante el uso de predicción espacial con respecto a las muestras de referencia en bloques cercanos de la misma imagen. Los bloques de video en un segmento intercodificada (P o B) de la imagen pueden usar predicción espacial con respecto a las muestras de referencia en bloques cercanos de la misma imagen o predicción temporal con respecto a las muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse fotogramas, y las imágenes de referencia pueden denominarse fotogramas de referencia.

SÍNTESIS

[0005] En general, esta divulgación describe técnicas relacionadas para la interpredicción en códecs de video. Más específicamente, esta divulgación describe dispositivos y técnicas relacionadas al refinamiento del vector de movimiento lateral del decodificador y a la terminación temprana de un proceso de búsqueda iterativo para determinar y/o refinar un vector de movimiento. Un codificador/decodificador de video se puede configurar para determinar un costo mínimo a través de una iteración de búsqueda anterior y actualizar un costo mínimo general a través de una iteración de búsqueda actual. El proceso de búsqueda iterativo luego puede terminarse temprano en función de una comparación del costo mínimo a través de la iteración de búsqueda previa y el costo mínimo general a través de la iteración de búsqueda actual. De esta manera, el proceso de búsqueda iterativo puede terminarse temprano cuando se observan mejoras insignificantes entre iteraciones de búsqueda sucesivas. En consecuencia, las técnicas de la presente divulgación pueden reducir la complejidad computacional sin aumentar la distorsión de forma material.

[0006] En un ejemplo, esta divulgación describe un método para decodificar datos de video, en donde el método comprende determinar un vector de movimiento para un bloque de datos de video mediante el uso de un proceso de derivación del vector de movimiento del lado del decodificador que incluye un proceso de búsqueda iterativa, en donde el proceso de búsqueda iterativa incluye: determinar un costo mínimo a través de una iteración de búsqueda anterior; actualizar el costo mínimo general a través de una iteración de búsqueda actual; y terminar anticipadamente el proceso de

búsqueda iterativa en función de una comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, y decodificar el bloque de datos de video mediante el uso del vector de movimiento.

[0007] En otro ejemplo, esta divulgación describe un método para codificar datos de video, en donde el método comprende determinar un vector de movimiento para un bloque de datos de video mediante el uso de un proceso de derivación del vector de movimiento del lado del decodificador que incluye un proceso de búsqueda iterativa, en donde el proceso de búsqueda iterativa incluye: determinar un costo mínimo a través de una iteración de búsqueda anterior; actualizar el costo mínimo general a través de una iteración de búsqueda actual; y terminar anticipadamente el proceso de búsqueda iterativa en función de una comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, y codificar el bloque de datos de video mediante el uso del vector de movimiento.

[0008] En otro ejemplo, un dispositivo para decodificar datos de video incluye una memoria configurada para almacenar los datos de video, y una circuitería de procesamiento configurada para determinar un vector de movimiento para un bloque de los datos de video utilizando un proceso de derivación del vector de movimiento del lado del decodificador que incluye un proceso de búsqueda iterativa, en donde el proceso de búsqueda iterativa incluye: determinar un costo mínimo a través de una iteración de búsqueda anterior; actualizar el costo mínimo general a través de una iteración de búsqueda actual; y terminar anticipadamente el proceso de búsqueda iterativa en función de una comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, y decodificar el bloque de datos de video mediante el uso del vector de movimiento.

[0009] En otro ejemplo, un dispositivo para codificar datos de video incluye una memoria configurada para almacenar los datos de video, y una circuitería de procesamiento configurada para determinar un vector de movimiento para un bloque de los datos de video utilizando un proceso de derivación del vector de movimiento del lado del decodificador que incluye un proceso de búsqueda iterativa, en donde el proceso de búsqueda iterativa incluye: determinar un costo mínimo a través de una iteración de búsqueda anterior; actualizar el costo mínimo general a través de una iteración de búsqueda actual; y terminar anticipadamente el proceso de búsqueda

iterativa en función de una comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, y codificar el bloque de datos de video mediante el uso del vector de movimiento.

[0010] Los detalles de uno o más ejemplos se exponen en las figuras adjuntas y en la descripción siguiente. Otras características, objetivos y ventajas serán evidentes a partir de la descripción, figuras y reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0011] La Figura 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de video de ejemplo que puede llevar a cabo las técnicas de esta divulgación.

[0012] La Figura 2 ilustra posiciones de ejemplo de candidatos de fusión espacial.

[0013] La Figura 3 ilustra un ejemplo de escalado de vector de movimiento para un candidato de fusión temporal.

[0014] La Figura 4 ilustra posiciones de candidatos de ejemplo para los candidatos de fusión temporal.

[0015] La Figura 5 ilustra un ejemplo de comparación bilateral.

[0016] La Figura 6 ilustra un ejemplo de una región de unidad de codificación extendida utilizada en el flujo óptico bidireccional.

[0017] La Figura 7 ilustra una región de búsqueda de diamante de ejemplo para el refinamiento de movimiento lateral del decodificador de multipase.

[0018] La Figura 8 ilustra un ejemplo de comparación de plantillas llevado a cabo en un área de búsqueda alrededor de un vector de movimiento inicial.

[0019] La Figura 9 es un diagrama de bloques que ilustra un codificador de video de ejemplo que puede realizar las técnicas de esta divulgación.

[0020] La Figura 10 es un diagrama de bloques que ilustra un decodificador de video de ejemplo que puede realizar las técnicas de esta divulgación.

[0021] La Figura 11 es un diagrama de flujo que ilustra un método de ejemplo para codificar un bloque actual de acuerdo con las técnicas de esta divulgación.

[0022] La Figura 12 es un diagrama de flujo que ilustra un método de ejemplo para decodificar un bloque actual de acuerdo con las técnicas de esta divulgación.

[0023] La Figura 13 es un diagrama de flujo que ilustra un método de ejemplo para terminar temprano una búsqueda iterativa de acuerdo con un ejemplo de la divulgación.

[0024] La Figura 14 es un diagrama de flujo que ilustra otro método de ejemplo para

decodificar un bloque actual de acuerdo con las técnicas de esta divulgación.

[0025] La Figura 15 es un diagrama de flujo que ilustra otro método de ejemplo para codificar un bloque actual de acuerdo con las técnicas de esta divulgación.

DESCRIPCIÓN DETALLADA

[0026] En algunas técnicas de interpredicción de ejemplo, se puede configurar un decodificador para llevar a cabo un proceso de búsqueda iterativa para determinar y/o refinar un vector de movimiento. Algunas técnicas de ejemplo incluyen el refinamiento del vector de movimiento lateral del decodificador (DMVR) de comparación bilateral y comparación de plantillas. En un ejemplo, el proceso de búsqueda de movimiento lateral del decodificador, para cada comparación bilateral en función del DMVR o la comparación de plantillas, se define un criterio de costo y se utiliza para calcular los costos para las posiciones de los candidatos dentro de un área de búsqueda predefinida. La información de movimiento asociada con la posición de candidatos que tenga el costo mínimo puede ser el resultado del proceso de búsqueda iterativa. El proceso de búsqueda se puede conducir en múltiples iteraciones. En cada iteración, se evalúa un subconjunto de posiciones de candidatos dentro de un área de búsqueda. Generalmente, la distancia entre la posición inicial y una posición de candidato aumenta tras cada iteración. En otras palabras, las posiciones de candidatos en la iteración de búsqueda actual pueden estar más alejada del punto inicial que la posición de la iteración anterior.

[0027] Una variable minCost puede representar el costo mínimo calculado durante el proceso de búsqueda iterativa y minCost puede representar el costo mínimo del proceso de búsqueda iterativa determinado hasta el momento durante el proceso de búsqueda iterativa. Se pueden aplicar algunos métodos de terminación anticipada para poner fin a un proceso de búsqueda. En algunos ejemplos, el proceso de búsqueda se puede terminar si el minCost es menor que un umbral, o el proceso de búsqueda se puede terminar si la mejor posición de candidato con el costo mínimo no cambió de un subconjunto al siguiente (por ejemplo, de una iteración a otra).

[0028] Sin embargo, en algunos ejemplos, el minCost calculado durante el proceso de búsqueda iterativa puede permanecer superior que un umbral sin disminuir a una velocidad para caer por debajo del umbral rápidamente. Por lo tanto, comparar simplemente el minCost con un umbral puede conducir a ejemplos de tiempos de decodificación más largos y mayor complejidad computacional de derivación de vector de movimiento lateral del decodificador y que a la vez tienen mejoras insignificantes de distorsión de imagen.

[0029] Para reducir aún más la complejidad computacional y mejorar la eficacia de la derivación del vector de movimiento lateral del decodificador, un codificador/decodificador de video se puede configurar para determinar un costo mínimo a través de una iteración de búsqueda anterior y actualizar un costo mínimo general a través de una iteración de búsqueda actual. En otras palabras, un codificador/decodificador de video puede determinar un costo mínimo a través de una iteración de búsqueda anterior y un costo mínimo general a través de una iteración de búsqueda actual. El proceso de búsqueda iterativo luego puede terminarse temprano en función de una comparación del costo mínimo a través de la iteración de búsqueda previa y el costo mínimo general a través de la iteración de búsqueda actual. En algunos ejemplos, determinar un costo mínimo a través de una iteración de búsqueda anterior puede incluir determinar un costo mínimo de todas las iteraciones de búsqueda anteriores. En algunos ejemplos, terminar el proceso de búsqueda iterativa temprano puede significar que el proceso de búsqueda iterativa divulgado en la presente puede tener una habilidad de terminarse antes de una terminación normal del proceso de búsqueda iterativa (es decir, el proceso de búsqueda iterativa se termina en función de una comparación de un costo mínimo (por ejemplo, minCost como se describe anteriormente) a través de la iteración de búsqueda iterativa y un umbral). Terminar el proceso de búsqueda iterativo al comparar el costo mínimo a través de la iteración de búsqueda previa y el costo mínimo general a través de la iteración de búsqueda actual puede reducir la complejidad computacional sin aumentar la distorsión en forma material.

[0030] En algunos ejemplos, el proceso de búsqueda iterativo luego puede terminarse temprano en función de una diferencia del costo mínimo a través de la iteración de búsqueda previa y el costo mínimo general a través de la iteración de búsqueda actual en relación con un umbral de diferencia. En algunos ejemplos, si la diferencia es menor que el umbral de diferencia, se da por terminado el proceso de búsqueda. En algunos ejemplos, el proceso de búsqueda iterativo luego puede terminarse temprano en función de una relación del costo mínimo a través de la iteración de búsqueda previa al costo mínimo general a través de la iteración de búsqueda actual en relación con un umbral de relación. En algunos ejemplos, el proceso de búsqueda iterativa puede terminar anticipadamente en función de una relación entre el costo mínimo de la iteración de búsqueda anterior y la diferencia entre el costo mínimo de la iteración de búsqueda anterior y el costo mínimo global de la iteración de búsqueda actual en relación con un umbral de relación. En algunos ejemplos, el umbral puede ser en

función de una característica del bloque codificado, tal como el tamaño del bloque codificado.

[0031] De esta manera, el proceso de búsqueda iterativo puede terminarse temprano cuando solo se observan mejoras insignificantes entre iteraciones de búsqueda sucesivas. Al terminar un proceso de búsqueda iterativa temprano mediante el uso de las técnicas de terminación temprana que se describen anteriormente, la complejidad computacional de la derivación del vector de movimiento lateral del decodificador puede reducirse sin aumentar la distorsión de forma material.

[0032] Las técnicas de terminación temprana que se describen anteriormente se pueden usar con cualquiera de las técnicas de derivación del vector de movimiento lateral del decodificador que incluyen un proceso de búsqueda con más de dos iteraciones. En un ejemplo, las técnicas de terminación temprana que se describen anteriormente pueden aplicarse en la comparación bilateral de DMVR o en la comparación de plantillas. En otros ejemplos, las técnicas de terminación temprana se describen anteriormente también se pueden utilizar junto con otros métodos de terminación temprana.

[0033] La Figura 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de video 100 de ejemplo que puede realizar las técnicas de esta divulgación. Las técnicas de esta divulgación se dirigen generalmente a encriptar (codificar y/o decodificar) datos de video. En general, los datos de video incluyen cualquier dato para procesar un video. Por lo tanto, los datos de video pueden incluir video sin procesar, no codificado, video codificado, video decodificado (por ejemplo, reconstruido) y metadatos de video, tales como datos de señalización.

[0034] Como se muestra en la Figura 1, el sistema 100 incluye un dispositivo fuente 102 que proporciona datos de video codificados que se desean decodificar y mostrar posteriormente mediante un dispositivo de destino 116, en este ejemplo. En particular, el dispositivo fuente 102 proporciona los datos de video al dispositivo de destino 116 a través de un medio legible por computadora 110. El dispositivo fuente 102 o el dispositivo de destino 116 puede comprender cualquiera de un amplio rango de dispositivos, lo que incluye computadoras de escritorio, computadoras portátiles (por ejemplo, laptop), dispositivos móviles, computadoras tipo tablet, decodificadores, aparatos telefónicos, tales como teléfonos inteligentes, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de juegos de video, dispositivo de streaming de video, dispositivos receptores de transmisión, o similar. En algunos casos, el dispositivo fuente 102 y el dispositivo de destino 116

pueden estar equipados para comunicación inalámbrica, y, por lo tanto, pueden denominarse dispositivos de comunicación inalámbrica.

[0035] En el ejemplo de la Figura 1, el dispositivo fuente 102 incluye la fuente de video 104, la memoria 106, el codificador de video 200 y la interfaz de salida 108. El dispositivo de destino 116 incluye la interfaz de entrada 122, el decodificador de video 300, la memoria 120 y el dispositivo de visualización 118. De acuerdo con esta divulgación, el codificador de video 200 del dispositivo fuente 102 y el decodificador de video 300 del dispositivo de destino 116 pueden configurarse para aplicar las técnicas de derivación del vector de movimiento del lado del decodificador. Por lo tanto, el dispositivo fuente 102 representa un ejemplo de un dispositivo de codificación de video, mientras que el dispositivo de destino 116 representa un ejemplo de un dispositivo de decodificación de video. En otros ejemplos, un dispositivo fuente y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo fuente 102 puede recibir datos de video de una fuente de video externa, tal como una cámara externa. Asimismo, el dispositivo de destino 116 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

[0036] El sistema 100 como se muestra en la Figura 1 es solo un ejemplo. En general, cualquier dispositivo de codificación y/o decodificación de video digital puede realizar técnicas para la derivación del vector de movimiento del lado del decodificador. El dispositivo fuente 102 y el dispositivo de destino 116 son solo ejemplos de los dispositivos de codificación, en donde el dispositivo fuente 102 genera datos de video codificados para la transmisión al dispositivo de destino 116. Esta divulgación se refiere a un dispositivo de "codificación" como un dispositivo que realiza la codificación (codificación y/o decodificación) de datos. Por lo tanto, el codificador de video 200 y el decodificador de video 300 representan ejemplos de dispositivos de codificación, en particular, un codificador de video y un decodificador de video, respectivamente. En algunos ejemplos, el dispositivo fuente 102 y de destino 116 pueden operar de manera sustancialmente simétrica, de modo que cada uno del dispositivo fuente 102 y el dispositivo de dispositivo de destino 116 incluya componentes de codificación y decodificación de video. Por lo tanto, el sistema 100 puede admitir la transmisión de video unidireccional o bidireccional entre el dispositivo fuente 102 y el dispositivo de destino 116, por ejemplo, para *streaming* de video, repetición de video, transmisión de video o videotelefonía.

[0037] En general, la fuente de video 104 representa una fuente de datos de video

(por ejemplo, datos de video sin procesar, no codificados) y proporciona una serie secuencial de imágenes (también denominadas "cuadros") de los datos de video al codificador de video 200, que codifica los datos para las imágenes. La fuente de video 104 del dispositivo fuente 102 puede incluir un dispositivo de captura de video, tal como una cámara de video, un archivo de video que contiene video sin procesar capturado anteriormente y/o una interfaz de transmisión de video para recibir videos de un proveedor de contenidos de video. Como alternativa adicional, la fuente de video 104 puede generar datos basados en gráficos informáticos como el video fuente, o una combinación de video en vivo, video archivado y video generado por computadora. En cada caso, el codificador de video 200 codifica los datos de video capturado, precapturado o generado por computadora. El codificador de video 200 puede reorganizar las imágenes del orden recibido (a veces denominado "orden de visualización") en un orden de codificación para la codificación. El codificador de video 200 puede generar un flujo de bits que incluye datos de video codificados. El dispositivo fuente 102 puede emitir entonces los datos de video codificados a través de la interfaz de salida 108 en un medio legible por computadora 110 para su recepción y/o recuperación mediante, por ejemplo, la interfaz de entrada 122 del dispositivo de destino 116.

[0038] La memoria 106 del dispositivo fuente 102 y la memoria 120 del dispositivo de destino 116 representan memorias de propósito general. En algunos ejemplos, las memorias 106, 120 pueden almacenar datos de video sin procesar, por ejemplo, video sin procesar de la fuente de video 104 y datos de video sin procesar, decodificados del decodificador de video 300. Adicional o alternativamente, las memorias 106, 120 pueden almacenar instrucciones de *software* ejecutables por, por ejemplo, el codificador de video 200 y el decodificador de video 300, respectivamente. Aunque la memoria 106 y la memoria 120 se muestran por separado del codificador de video 200 y el decodificador de video 300 en este ejemplo, debe entenderse que el codificador de video 200 y el decodificador de video 300 también pueden incluir memorias internas para propósitos funcionalmente similares o equivalentes. Además, las memorias 106, 120 pueden almacenar datos de video codificados, por ejemplo, la salida del codificador de video 200 y la entrada al decodificador de video 300. En algunos ejemplos, los segmentos de memorias 106, 120 se pueden ubicar como uno o más búfer de video, por ejemplo, para almacenar datos de video sin procesar, decodificados o codificados.

[0039] El medio legible por computadora 110 puede representar cualquier tipo de

medio o dispositivo capaz de transportar los datos de video codificados desde el dispositivo fuente 102 al dispositivo de destino 116. En un ejemplo, el medio legible por computadora 110 representa un medio de comunicación para permitir que el dispositivo fuente 102 transmita datos de video codificados directamente al dispositivo de destino 116 en tiempo real, por ejemplo, a través de una red de radiofrecuencia o una red basada en computadora. La interfaz de salida 108 puede modular una señal de transmisión que incluye los datos de video codificados, y la interfaz de entrada 122 puede demodular la señal de transmisión recibida, de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrica o por cable, tales como un espectro de frecuencia de radio (RF) o una o más líneas de transmisión física. El medio de comunicación 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 medio de comunicación puede incluir enrutadores, conmutadores, estaciones base, o cualquier otro equipo que pueda ser útil para facilitar la comunicación del dispositivo fuente 102 al dispositivo de destino 116.

[0040] En algunos ejemplos, el dispositivo fuente 102 puede enviar datos codificados desde la interfaz de salida 108 al dispositivo de almacenamiento 112. De manera similar, el dispositivo de destino 116 puede acceder a datos codificados desde el dispositivo de almacenamiento 112 a través de la interfaz de entrada 122. El dispositivo de almacenamiento 112 puede incluir cualquiera de una variedad de medios de almacenamiento de datos distribuidos o a los que se accede localmente, tales como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil, o cualquier otro medio de almacenamiento digital adecuado para el almacenamiento de datos de video codificados.

[0041] En algunos ejemplos, el dispositivo fuente 102 puede enviar datos de video codificados al servidor de archivos 114 u otro dispositivo de almacenamiento intermedio que puede almacenar los datos de video codificados generados por el dispositivo fuente 102. El dispositivo de destino 116 puede acceder a los datos de video almacenados desde el servidor de archivos 114 mediante *streaming* o descarga.

[0042] El servidor de archivos 114 puede ser cualquier tipo de dispositivo de servidor capaz de almacenar datos de video y de transmitir los datos de video codificados al dispositivo de destino 116. El servidor de archivos 114 puede representar un servidor web (por ejemplo, para un sitio web), un servidor configurado para proporcionar un servicio de protocolo de transferencia de archivos (tal como el protocolo de

transferencia de archivos (FTP) o el protocolo de entrega de archivos a través de un objeto de transporte unidireccional (FLUTE)), un dispositivo de red de entrega de contenido (CDN), un servidor de protocolo de transferencia de hipertexto (HTTP), un servidor de servicio de multidifusión de emisión multimedia (MBMS) o MBMS mejorado (eMBMS) y/o un dispositivo de almacenamiento conectado a la red (NAS). El servidor de archivos 114 puede, de manera adicional o alternativa, implementar uno o más protocolos de transmisión HTTP, tal como transmisión dinámica adaptativa a través de HTTP (DASH), transmisión en vivo de HTTP (HLS), protocolo de transmisión en tiempo real (RTSP), transmisión dinámica de HTTP o similar.

[0043] El dispositivo de destino 116 puede acceder a los datos de video codificados del servidor de archivos 114 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 por cable (por ejemplo, línea de abonado digital (DSL, cable módem, etc.), o una combinación de ambos que es adecuada para acceder a los datos de video codificados y almacenados en el servidor de archivos 114. La interfaz de entrada 122 puede configurarse para funcionar de acuerdo con uno o más de los diversos protocolos analizados anteriormente para recuperar o recibir datos de medios a partir del servidor de archivos 114, u otros protocolos para la recuperación de datos de medios.

[0044] La interfaz de salida 108 y la interfaz de entrada 122 pueden representar transmisores/receptores inalámbricos, módems, componentes de red cableados (por ejemplo, tarjetas Ethernet), componentes de comunicación inalámbrica que operan de acuerdo con cualquiera de una variedad de estándares IEEE 802.11 u otros componentes físicos. En los ejemplos en los que la interfaz de salida 108 y la interfaz de entrada 122 comprenden componentes inalámbricos, la interfaz de salida 108 y la interfaz de entrada 122 pueden configurarse para transferir datos, tales como datos de video codificados, de acuerdo con un estándar de comunicación celular, tal como 4G, 4G-LTE (evolución a largo plazo), LTE avanzada, 5G o similar. En algunos ejemplos en los que la interfaz de salida 108 comprende un transmisor inalámbrico, la interfaz de salida 108 y la interfaz de entrada 122 pueden configurarse para transferir datos, tales como datos de video codificados, de acuerdo con otros estándares inalámbricos, tales como una especificación IEEE 802.11, una especificación IEEE 802.15 (por ejemplo, ZigBee™), un estándar Bluetooth™ o similar. En algunos ejemplos, el dispositivo fuente 102 y/o el dispositivo de destino 116 pueden incluir respectivos dispositivos de sistema en un chip (SoC). Por ejemplo, el dispositivo fuente 102 puede

incluir un dispositivo de SoC para realizar la funcionalidad atribuida al codificador de video 200 y/o la interfaz de salida 108, y el dispositivo de destino 116 puede incluir un dispositivo de SoC para realizar la funcionalidad atribuida al decodificador de video 300 y/o la interfaz de entrada 122.

[0045] Las técnicas de esta divulgación se pueden aplicar a codificación de video para admitir diversas aplicaciones multimedia, tales como transmisiones por televisión inalámbrica, transmisiones por televisión por cable, transmisiones por televisión satelital, transmisiones de video por *streaming* en Internet, tales como *streaming* adaptativo dinámico en HTTP (DASH), video digital que se codifica en un medio de almacenamiento de datos, decodificación de video digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones.

[0046] La interfaz de entrada 122 del dispositivo de destino 116 recibe un flujo de bits de video codificado desde un medio legible por computadora 110 (por ejemplo, un medio de comunicación, el dispositivo de almacenamiento 112, el servidor de archivos 114 o similar). El flujo de bits de video codificado puede incluir información de señalización definida por el codificador de video 200, que también es utilizada por el decodificador de video 300, tal como elementos de sintaxis que tienen valores que describen características y/o procesamiento de bloques de video u otras unidades codificadas (por ejemplo, segmentos, imágenes, grupos de imágenes, secuencias o similar). El dispositivo de visualización 118 muestra imágenes decodificadas de los datos de video decodificados a un usuario. El dispositivo de visualización 118 puede representar cualquiera de los diversos dispositivos de visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodo orgánico de emisión de luz (OLED), u otro tipo de dispositivo de visualización.

[0047] Aunque no se muestra en la Figura 1, en algunos ejemplos, el codificador de video 200 y el decodificador de video 300 pueden cada uno estar integrados con un codificador de audio y/o un decodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro *hardware* y/o *software*, para manejar flujos múltiples, incluso la codificación de audio y video en un flujo común de datos.

[0048] El codificador de video 200 y el decodificador de video 300 pueden implementarse como cualquiera de una variedad de circuitería codificadora y/o decodificadora adecuada, tal como uno o más microprocesadores, procesadores de señal digital (DSP), circuitos integrados de aplicación específica (ASIC), matrices de puertas programables en campo (FPGA), circuitos discretos lógicos, *software*, *hardware*, *firmware* o cualquier combinación de estos. Cuando las técnicas se

implementan parcialmente en *software*, un dispositivo puede almacenar las instrucciones para el *software* en un medio legible por computadora no transitorio adecuado y ejecutar las instrucciones en *hardware* mediante el uso de uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno del codificador de video 200 y el decodificador de video 300 pueden incluirse en uno o más codificadores o decodificadores, cualquiera de los cuales pueden estar integrados como parte de un codificador/decodificador combinado (CODEC) en un dispositivo respectivo. Un dispositivo que incluye un codificador de video 200 y/o un decodificador de video 300 puede comprender un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrica, tal como un teléfono celular.

[0049] El codificador de video 200 y el decodificador de video 300 pueden operar de acuerdo con un estándar de codificación de video, tal como ITU-T H.265, también conocido como codificación de video de alta eficiencia (HEVC) o extensiones de este, tales como las extensiones de codificación de video multivisión y/o escalable. Alternativamente, el codificador de video 200 y el decodificador de video 300 pueden operar de acuerdo con otros estándares patentados o de la industria, tales como ITU-T H.266, también denominado codificación de video versátil (VVC). En otros ejemplos, el codificador de video 200 y el decodificador de video 300 pueden operar de acuerdo con un códec/formato de video patentado, como *AOMedia Video 1* (AV1), extensiones de AV1 y/o versiones sucesoras de AV1 (por ejemplo, AV2). En otros ejemplos, el codificador de video 200 y el decodificador de video 300 pueden operar de acuerdo con otros formatos patentados o estándares de la industria. Las técnicas de esta divulgación, sin embargo, no se limitan a ningún estándar de codificación o formato particular. En general, el codificador de video 200 y el decodificador de video 300 pueden configurarse para llevar a cabo las técnicas de esta divulgación junto con cualquier técnica de codificación de video que utilice las técnicas de derivación del vector de movimiento lateral del decodificador.

[0050] En general, el codificador de video 200 y el decodificador de video 300 pueden realizar una codificación de imágenes basada en bloques. El término "bloque" generalmente se refiere a una estructura que incluye datos para ser procesados (por ejemplo, codificados, decodificados o usados de otro modo en el proceso de codificación y/o decodificación). Por ejemplo, un bloque puede incluir una matriz bidimensional de muestras de datos de luminancia y/o crominancia. En general, el codificador de video 200 y el decodificador de video 300 pueden codificar datos de video representados en un formato YUV (por ejemplo, Y, Cb, Cr). Es decir, en lugar de

codificar datos de rojo, verde y azul (RGB) para muestras de una imagen, el codificador de video 200 y el decodificador de video 300 pueden codificar componentes de luminancia y crominancia, en donde los componentes de crominancia pueden incluir componentes de crominancia de tono rojo y azul. En algunos ejemplos, el codificador de video 200 convierte los datos con formato RGB recibidos en una representación YUV antes de la codificación, y el decodificador de video 300 convierte la representación YUV en el formato RGB. Alternativamente, las unidades de procesamiento previo y posterior (no mostradas) pueden realizar estas conversiones.

[0051] Esta divulgación puede referirse generalmente a la codificación (por ejemplo, codificación y decodificación) de imágenes para incluir el proceso de codificación o decodificación de datos de la imagen. De manera similar, esta divulgación puede referirse a la codificación de bloques de una imagen para incluir el proceso de codificación o decodificación de datos para los bloques, por ejemplo, predicción y/o codificación residual. Un flujo de bits de video codificado generalmente incluye una serie de valores para elementos de sintaxis representativos de decisiones de codificación (por ejemplo, modos de codificación) y partición de imágenes en bloques. Por tanto, las referencias a la codificación de una imagen o un bloque deben entenderse generalmente como valores de codificación para los elementos de sintaxis que forman la imagen o el bloque.

[0052] HEVC define diversos bloques, incluso las unidades de codificación (CU), las unidades de predicción (PU) y las unidades de transformación (TU). De acuerdo con HEVC, un encriptador de video (tal como el codificador de video 200) divide una unidad de árbol de codificación (CTU) en CU de acuerdo con una estructura de árbol cuaternario. Es decir, el encriptador de video divide las CTU y CU en cuatro cuadrados iguales que no se superponen, y cada nodo del árbol cuaternario tiene cero o cuatro nodos secundarios. Los nodos sin nodos secundarios pueden denominarse "nodos hoja", y las CU de dichos nodos hoja pueden incluir una o más PU y/o una o más TU. El encriptador de video puede dividir además las PU y las TU. Por ejemplo, en HEVC, un árbol cuaternario residual (RQT) representa la partición de TU. En HEVC, las PU representan datos de interpredicción, mientras que las TU representan datos residuales. Las CU que se interpredicen incluyen información de intrapredicción, tal como una indicación intramodo.

[0053] Como otro ejemplo, el codificador de video 200 y el decodificador de video 300 pueden configurarse para operar de acuerdo con VVC. De acuerdo con VVC, un codificador de video (tal como el codificador de video 200) divide una imagen en una

pluralidad de unidades de árbol de codificación (CTU). El codificador de video 200 puede dividir una CTU de acuerdo con una estructura de árbol, tal como una estructura de árbol binario de árbol cuaternario (QTBT) o una estructura de árbol de tipos múltiples (MTT). La estructura QTBT elimina los conceptos de múltiples tipos de partición, tales como la separación entre CU, PU y TU de HEVC. Una estructura QTBT incluye dos niveles: un primer nivel dividido de acuerdo con la partición de árbol cuaternario y un segundo nivel dividido de acuerdo con la partición de árbol binario. Un nodo raíz de la estructura QTBT corresponde a una CTU. Los nodos hoja de los árboles binarios corresponden a unidades de codificación (CU).

[0054] En una estructura de partición MTT, los bloques se pueden particionar mediante el uso de una partición de árbol cuaternario (QT), una partición de árbol binario (BT) y uno o más tipos de particiones de árbol triple (TT) (también denominado árbol ternario). Una partición de árbol triple o ternario es una partición en la que un bloque se divide en tres subbloques. En algunos ejemplos, una partición de árbol triple o ternario divide un bloque en tres subbloques sin dividir el bloque original por el centro. Los tipos de partición en MTT (por ejemplo, QT, BT y TT) pueden ser simétricos o asimétricos.

[0055] Cuando se opera de acuerdo con el códec AV1, el dispositivo de codificación 200 y el decodificador de video 300 pueden configurarse para codificar datos de video en bloques. En AV1, el bloque de codificación más grande que se puede procesar se denomina superbloque. En AV1, un superbloque puede tener muestras de luminancia de 128x128 o muestras de luminancia de 64x64. Sin embargo, en los formatos de codificación de video posteriores sucesores (por ejemplo, AV2), un superbloque puede estar definido por diferentes (por ejemplo, mayores) tamaños de muestras de luminancia. En algunos ejemplos, un superbloque es el nivel superior de un árbol cuaternario de bloques. El codificador de video 200 además puede dividir un superbloque en bloques de codificación más pequeños. El codificador de video 200 puede dividir un superbloque y otros bloques de codificación en bloques más pequeños mediante el uso de partición cuadrada o no cuadrada. Los bloques no cuadrados pueden incluir bloques $N/2 \times N$, $N \times N/2$, $N/4 \times N$ y $N \times N/4$. El codificador de video 200 y el decodificador de video 300 pueden llevar a cabo procesos de predicción y transformación separados en cada uno de los bloques de codificación.

[0056] AV1 también define un mosaico de datos de video. Un mosaico es una matriz rectangular de superbloques que se pueden codificar independientemente de otros mosaicos. Es decir, el codificador de video 200 y el decodificador de video 300 pueden

codificar y decodificar, respectivamente, bloques de codificación dentro de un mosaico sin usar datos de video de otros mosaicos. Sin embargo, el codificador de video 200 y el decodificador de video 300 pueden llevar a cabo un filtrado a través de los límites de los mosaicos. Los mosaicos pueden tener tamaño uniforme o no uniforme. La codificación basada en mosaicos puede permitir el procesamiento paralelo y/o multitenhebrado para implementaciones de codificador y decodificador.

[0057] En algunos ejemplos, el codificador de video 200 y el decodificador de video 300 pueden usar una sola estructura QTBT o MTT para representar cada uno de los componentes de luminancia y crominancia, mientras que en otros ejemplos, el codificador de video 200 y el decodificador de video 300 pueden usar dos o más estructuras QTBT o MTT, como una estructura QTBT/MTT para el componente de luminancia y otra estructura QTBT/MTT para ambos componentes de crominancia (o dos estructuras QTBT/MTT para los respectivos componentes de crominancia).

[0058] El codificador de video 200 y el decodificador de video 300 pueden configurarse para usar partición de árbol cuaternario, partición QTBT, partición MTT, partición de super bloque u otras estructuras de partición.

[0059] En algunos ejemplos, una CTU incluye un bloque de árbol de codificación (CTB) de muestras de luminancia, dos CTB de muestras de crominancia correspondientes de una imagen que tiene tres matrices de muestra, o un CTB de muestras de una imagen monocromática o una imagen que se codifica mediante el uso de tres planos de color separados y estructuras sintácticas utilizadas para codificar las muestras. Un CTB puede ser un bloque $N \times N$ de muestras para algún valor de N tal que la división de un componente en CTB es una partición. Un componente es una matriz o muestra única de una de las tres matrices (luminancia y dos crominancia) que componen una imagen en formato de color 4:2:0, 4:2:2 o 4:4:4 o la matriz o una muestra única de la matriz que compone una imagen en formato monocromático. En algunos ejemplos, un bloque de codificación es un bloque $M \times N$ de muestras para algunos valores de M y N , de modo que la división de un CTB en bloques de codificación es una partición.

[0060] Los bloques (por ejemplo, CTU o CU) pueden agruparse en diversas formas en una imagen. A modo de ejemplo, un ladrillo puede referirse a una región rectangular de filas de CTU dentro de un mosaico particular en una imagen. Un mosaico puede ser una región rectangular de CTB dentro de una columna de mosaicos en particular y una fila de mosaicos en particular en una imagen. Una columna de mosaico se refiere a una región rectangular de CTU que tiene una altura igual a la altura de la imagen y un

ancho especificado por los elementos de sintaxis (por ejemplo, tal como en un conjunto de parámetros de imagen). Una fila de mosaico se refiere a una región rectangular de CTU que tiene una altura especificada por los elementos de sintaxis (por ejemplo, tal como en un conjunto de parámetros de imagen) y un ancho igual al ancho de la imagen.

[0061] En algunos ejemplos, un mosaico puede particionarse en múltiples ladrillos, cada uno de los cuales puede incluir una o más filas de CTU dentro del mosaico. Un mosaico que no está particionado en múltiples ladrillos también se puede denominar "ladrillo". Sin embargo, un ladrillo que es un subconjunto real de un mosaico no se puede denominar mosaico. Los ladrillos en una imagen se pueden disponer en un segmento. Un segmento puede ser un número entero de ladrillos de una imagen que pueden contenerse exclusivamente en una única unidad de capa de abstracción de red (NAL). En algunos ejemplos, un segmento incluye una cantidad de mosaicos completos o solo una secuencia consecutiva de ladrillos completos de un mosaico.

[0062] Esta divulgación puede usar "NxN" y "N por N" indistintamente para referirse a las dimensiones de muestra de un bloque (tal como una CU u otro bloque de video) en términos de dimensiones verticales y horizontales, por ejemplo, muestras de 16x16 o muestras de 16 por 16. En general, una CU de 16x16 tendrá 16 muestras en dirección vertical ($y = 16$) y 16 muestras en dirección horizontal ($x = 16$). Asimismo, una CU de NxN generalmente tiene N muestras en una dirección vertical y N muestras en una dirección horizontal, donde N representa un valor entero no negativo. Las muestras en una CU pueden disponerse en filas y columnas. Además, las CU no necesitan tener necesariamente la misma cantidad de muestras en la dirección horizontal que en la dirección vertical. Por ejemplo, las CU pueden comprender muestras NxM, en donde M no es necesariamente igual a N.

[0063] El codificador de video 200 codifica datos de video para CU que representan información de predicción y/o residual y otra información. La información de predicción indica cómo se va a predecir la CU para formar un bloque de predicción para la CU. La información residual generalmente representa diferencias muestra por muestra entre muestras de la CU antes de la codificación y el bloque de predicción.

[0064] Para predecir una CU, el codificador de video 200 puede formar generalmente un bloque de predicción para la CU mediante interpredicción o intrapredicción. La interpredicción generalmente se refiere a predecir la CU a partir de datos de una imagen previamente codificada, mientras que la intrapredicción generalmente se refiere a predecir la CU a partir de datos previamente codificados de la misma imagen.

Para realizar la interpredicción, el codificador de video 200 puede generar el bloque de predicción mediante el uso de uno o más vectores de movimiento. El codificador de video 200 generalmente puede realizar una búsqueda de movimiento para identificar un bloque de referencia que se asemeje mucho a la CU, por ejemplo, en términos de diferencias entre la CU y el bloque de referencia. El codificador de video 200 puede calcular una métrica de diferencia mediante el uso de una suma de diferencia absoluta (SAD), suma de diferencias cuadradas (SSD), media de diferencia absoluta (MAD), media de diferencias al cuadrado (MSD) u otros cálculos de diferencias similares para determinar si un bloque de referencia coincide estrechamente con la CU actual. En algunos ejemplos, el codificador de video 200 puede predecir la CU actual mediante el uso de predicción unidireccional o predicción bidireccional.

[0065] Algunos ejemplos de VVC también proporcionan un modo de compensación de movimiento afín, que puede considerarse un modo de interpredicción. En el modo de compensación de movimiento afín, el codificador de video 200 puede determinar dos o más vectores de movimiento que representan un movimiento no traslacional, tal como acercar o alejar, rotación, movimiento en perspectiva u otros tipos de movimiento irregular.

[0066] Para realizar la intrapredicción, el codificador de video 200 puede seleccionar un modo de intrapredicción para generar el bloque de predicción. Algunos ejemplos de VVC proporcionan sesenta y siete modos de intrapredicción, incluso diversos modos direccionales, así como el modo planar y el modo DC. En general, el codificador de video 200 selecciona un modo de intrapredicción que describe muestras cercanas a un bloque actual (por ejemplo, un bloque de una CU) desde el cual predecir muestras del bloque actual. Por lo general, dichas muestras pueden estar arriba, arriba y a la izquierda, o a la izquierda del bloque actual en la misma imagen que el bloque actual, asumiendo que el codificador de video 200 codifica CTU y CU en orden de exploración de trama (de izquierda a derecha, de arriba a abajo).

[0067] El codificador de video 200 codifica datos que representan el modo de predicción para un bloque actual. Por ejemplo, para los modos de interpredicción, el codificador de video 200 puede codificar datos que representen cuál de los diversos modos de interpredicción disponibles se usa, así como información de movimiento para el modo correspondiente. Para la interpredicción unidireccional o bidireccional, por ejemplo, el codificador de video 200 puede codificar vectores de movimiento mediante el uso de predicción avanzada de vectores de movimiento (AMVP) o modo de fusión. El codificador de video 200 puede usar modos similares para codificar

vectores de movimiento para el modo de compensación de movimiento afín.

[0068] AV1 incluye dos técnicas generales para codificar y decodificar un bloque de codificación de datos de video. Las dos técnicas generales son la intrapredicción (por ejemplo, predicción intracuadro o predicción espacial) y la interpredicción (por ejemplo, predicción intercuadro o predicción temporal). En el contexto de AV1, cuando se predicen bloques de un cuadro actual de datos de video mediante la utilización de un modo de codificación por intrapredicción, codificador de video 200 y el decodificador de video 300 no usan datos de video de otros cuadros de datos de video. Para la mayoría de los modos de intrapredicción, el codificador de video 200 codifica bloques de un cuadro actual en función de la diferencia entre los valores de muestra en el bloque actual y los valores predichos generados a partir de muestras de referencia en el mismo cuadro. El codificador de video 200 determina los valores previstos generados a partir de las muestras de referencia en función del modo de intrapredicción.

[0069] Después de la predicción, tal como intrapredicción o interpredicción de un bloque, el codificador de video 200 puede calcular datos residuales para el bloque. Los datos residuales, tales como un bloque residual, representan las diferencias muestra por muestra entre el bloque y un bloque de predicción para el bloque, formado mediante el uso del modo de predicción correspondiente. El codificador de video 200 puede aplicar una o más transformaciones al bloque residual, para producir datos transformados en un dominio de transformación en lugar del dominio de muestra. Por ejemplo, el codificador de video 200 puede aplicar una transformada de coseno discreta (DCT), una transformada de número entero, una transformada de ondículas o una transformada conceptualmente similar a los datos de video residuales. Además, el codificador de video 200 puede aplicar una transformada secundaria después de la primera transformada, tal como una transformada secundaria no separable dependiente del modo (MDNSST), una transformada dependiente de la señal, una transformada de Karhunen-Loeve (KLT) o similares. El codificador de video 200 produce coeficientes de transformada tras la aplicación de la una o más transformadas.

[0070] Como se indicó anteriormente, tras cualquier transformación para producir coeficientes de transformación, el codificador de video 200 puede llevar a cabo la cuantificación de los coeficientes de transformación. Cuantificación se refiere generalmente a un proceso donde los coeficientes de transformada se cuantifican para reducir posiblemente la cantidad de datos utilizados para representar los coeficientes

de transformada, lo que proporciona una compresión adicional. Al llevar a cabo el proceso de cuantificación, el codificador de video 200 puede reducir la profundidad de bits asociada con algunos o todos los coeficientes de transformación. Por ejemplo, el codificador de video 200 puede redondear un valor de n bits a un valor de m bits durante la cuantificación, donde n es mayor que m . En algunos ejemplos, para llevar a cabo la cuantificación, el codificador de video 200 puede llevar a cabo un desplazamiento a la derecha bit a bit del valor que se va a cuantificar.

[0071] Después de la cuantificación, el codificador de video 200 puede explorar los coeficientes de transformación, lo que produce un vector unidimensional a partir de la matriz bidimensional que incluye los coeficientes de transformación cuantificados. La exploración puede diseñarse para colocar coeficientes de transformación de energía más alta (y, por lo tanto, de frecuencia más baja) en la parte delantera del vector y para colocar coeficientes de transformación de energía más baja (y, por lo tanto, de frecuencia más alta) en la parte posterior del vector. En algunos ejemplos, el codificador de video 200 puede utilizar un orden de exploración predefinido para explorar los coeficientes de transformación cuantificados para producir un vector serializado, y luego codificar por entropía los coeficientes de transformación cuantificados del vector. En otros ejemplos, el codificador de video 200 puede realizar una exploración adaptativa. Después de explorar los coeficientes de transformación cuantificados para formar el vector unidimensional, el codificador de video 200 puede codificar por entropía el vector unidimensional, por ejemplo, de acuerdo con la codificación aritmética binaria adaptativa al contexto (CABAC). El codificador de video 200 también puede codificar por entropía valores para elementos sintácticos que describen metadatos asociados con los datos de video codificados para su uso por el decodificador de video 300 en la decodificación de los datos de video.

[0072] Para realizar CABAC, el codificador de video 200 puede asignar un contexto dentro de un modelo de contexto a un símbolo a transmitir. El contexto puede referirse, por ejemplo, a si los valores cercanos del símbolo tienen valor cero o no. La determinación de la probabilidad puede ser en función de un contexto asignado al símbolo.

[0073] El codificador de video 200 puede generar además datos de sintaxis, tales como datos de sintaxis basados en bloques, datos de sintaxis basados en imágenes y datos de sintaxis basados en secuencias, al decodificador de video 300, por ejemplo, en un encabezado de imagen, un encabezado de bloque, un encabezado de segmento, u otros datos de sintaxis, tales como un conjunto de parámetros de

secuencia (SPS), un conjunto de parámetros de imagen (PPS) o un conjunto de parámetros de video (VPS). El decodificador de video 300 puede decodificar igualmente tales datos de sintaxis para determinar cómo decodificar los datos de video correspondientes.

[0074] De esta manera, el codificador de video 200 puede generar un flujo de bits que incluye datos de video codificados, por ejemplo, elementos de sintaxis que describen la partición de una imagen en bloques (por ejemplo, CU) e información de predicción y/o residual para los bloques. Finalmente, el decodificador de video 300 puede recibir el flujo de bits y decodificar los datos de video codificados.

[0075] En general, el decodificador de video 300 lleva a cabo un proceso recíproco al llevado a cabo por el codificador de video 200 para decodificar los datos de video codificados del flujo de bits. Por ejemplo, el decodificador de video 300 puede decodificar valores para elementos de sintaxis del flujo de bits mediante el uso de CABAC de una manera sustancialmente similar, aunque recíproca al proceso de codificación CABAC del codificador de video 200. Los elementos de sintaxis pueden definir la información de partición para la partición de una imagen en CTU, y la partición de cada CTU de acuerdo con una estructura de partición correspondiente, tal como una estructura QTBT, para definir las CU de la CTU. Los elementos de sintaxis pueden definir además la información de predicción y residual para los bloques (por ejemplo, las CU) de datos de video.

[0076] La información residual puede representarse, por ejemplo, mediante coeficientes de transformación cuantificados. El decodificador de video 300 puede cuantificar de forma inversa y transformar de forma inversa los coeficientes de transformación cuantificados de un bloque para reproducir un bloque residual para el bloque. El decodificador de video 300 utiliza un modo de predicción señalado (intrapredicción o interpredicción) e información de predicción relacionada (por ejemplo, información de movimiento para la interpredicción) para formar un bloque de predicción para el bloque. El decodificador de video 300 puede combinar entonces el bloque de predicción y el bloque residual (muestra por muestra) para reproducir el bloque original. El decodificador de video 300 puede llevar a cabo un procesamiento adicional, tal como llevar a cabo un proceso de desbloqueo para reducir los artefactos visuales a lo largo de los límites del bloque.

[0077] Esta divulgación en general puede referirse a la "señalización" de cierta información, tal como elementos de sintaxis. El término "señalización" puede referirse generalmente a la comunicación de valores para elementos de sintaxis y/u otros datos

utilizados para decodificar datos de video codificados. Es decir, el codificador de video 200 puede señalar valores para elementos de sintaxis en el flujo de bits. En general, la señalización se refiere a generar un valor en el flujo de bits. Como se señaló anteriormente, el dispositivo fuente 102 puede transportar el flujo de bits al dispositivo de destino 116 sustancialmente en tiempo real, o no en tiempo real, como podría ocurrir cuando se almacenan elementos sintácticos en el dispositivo de almacenamiento 112 para su posterior recuperación por el dispositivo de destino 116.

[0078] Las técnicas de esta divulgación están relacionadas con la intrapredicción en códecs de video. Más específicamente, esta divulgación describe dispositivos y técnicas relacionadas al refinamiento del vector de movimiento lateral del decodificador mediante el uso de comparación bilateral. En consecuencia con las técnicas de la presente divulgación, como se describirá en mayor detalle a continuación, el decodificador de video 300 puede incluir una memoria configurada para almacenar los datos de video; y una circuitería de procesamiento que pueda configurarse para determinar un vector de movimiento para un bloque de datos de video utilizando un proceso de derivación del vector de movimiento del lado del descodificador que incluye un proceso de búsqueda iterativa, en donde el proceso de búsqueda iterativa incluye: determinar un costo mínimo a través de una iteración de búsqueda anterior; actualizar el costo mínimo general a través de una iteración de búsqueda actual; y terminar anticipadamente el proceso de búsqueda iterativa en función de una comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, y decodificar el bloque de datos de video mediante el uso del vector de movimiento.

[0079] En consecuencia con las técnicas de la presente divulgación, como se describirá en mayor detalle a continuación, el codificador de video 200 puede incluir una memoria configurada para almacenar los datos de video; y una circuitería de procesamiento que pueda configurarse para determinar un vector de movimiento para un bloque de datos de video utilizando un proceso de derivación del vector de movimiento del lado del descodificador que incluye un proceso de búsqueda iterativa, en donde el proceso de búsqueda iterativa incluye: determinar un costo mínimo a través de una iteración de búsqueda anterior; actualizar el costo mínimo general a través de una iteración de búsqueda actual; y terminar anticipadamente el proceso de búsqueda iterativa en función de una comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, y codificar el bloque de datos de video mediante el uso del vector de

movimiento.

[0080] Comparación bilateral

[0081] La comparación bilateral (BM) es una técnica que refina un par de vectores de movimiento iniciales MV0 y MV1. Generalmente, el BM incluye buscar alrededor de los vectores de movimiento MV0 y MV1 para derivar los MV refinados MV0' y MV1', que minimizan el costo de comparación de bloques. El costo de comparación de bloques mide la similitud entre los dos predictores compensados por movimiento generados mediante el uso de los dos MV (algunos criterios típicos son la suma de la diferencia absoluta (SAD), la suma de la diferencia absoluta transformada (SATD), la suma del error cuadrático (SSE), entre otros). El costo de comparación de bloques también puede incluir un término de regularización que se deriva en base a las diferencias de MV entre el MV actual y el MV inicial. También se pueden aplicar algunas limitaciones a las diferencias de MV entre MVD0 (MV0'-MV0) y MVD1 (MV1'-MV1). Generalmente, los codificadores de video se configuran con la suposición de que MVD0 y MVD1 deben ser proporcionales a las distancias temporales (TD) entre la imagen actual y las imágenes de referencia señaladas por los dos MV. Sin embargo, en algunas aplicaciones, el proceso de BM se puede diseñar con la suposición de que MVD0 es igual a - MVD1.

[0082] Interpretación en el estándar de codificación de video versátil

[0083] Para las CU interpredichas en VVC, los parámetros de movimiento incluyen vectores de movimiento, índices de imágenes de referencia, e índice de uso de la lista de imágenes de referencia, así como cualquier información adicional necesaria para la nueva característica de codificación de VVC. Los parámetros de movimiento se utilizan para la generación de muestras interpredichas. Los parámetros de movimiento pueden señalarse de forma explícita o implícita. Cuando una CU se codifica con el modo de salto, la CU se asocia a una PU y no tiene coeficientes residuales significativos, delta de vector de movimiento codificado o índice de imagen de referencia. Se puede especificar un modo de fusión mediante el cual los parámetros de movimiento para la CU actual se obtienen de las CU vecinas, lo que incluye candidatos espaciales y temporales, así como cronogramas adicionales introducidos en VVC. El modo de fusión puede aplicarse a cualquier CU interpredicha, no solo al modo de salto. Algunos otros modos de predicción aparte del modo de fusión incluyen la transmisión explícita de parámetros de movimiento, donde un vector de movimiento (por ejemplo, una diferencia de vector de movimiento), un índice de imágenes de referencia correspondiente para cada lista de imágenes de referencia y un marcador de uso de

lista de imágenes de referencia y otra información necesaria se señalizan explícitamente para cada CU.

[0084] Además de las funciones de intercodificación de HEVC, VVC incluye una serie de herramientas de codificación de interpredicción nuevas y refinadas, listadas de la manera siguiente:

- Predicción de fusión extendida
- Modo de fusión con MVD (MMVD)
- Señalización de MVD simétrico (SMVD)
- Predicción compensada de movimiento afín
- Predicción de vector de movimiento temporal basado en subbloques (SbTMVP)
- Resolución de vector de movimiento adaptivo (AMVR)
- Almacenamiento de campo de movimiento Almacenamiento de MV de muestra de luminancia de 1/16.º y compresión de campo de movimiento de 8x8
- Bipredicción con ponderación a nivel de CU (BCW)
- Flujo óptico bidireccional (BDOF)
- Refinamiento de vector de movimiento lateral del decodificador (DMVR)
- Modo de partición geométrico (GPM)
- Interpredicción e intrapredicción combinada (CIIP)

[0085] Predicción de fusión extendida

[0086] En el modo de fusión regular VVC, la lista de candidatos de fusión construye al incluir los siguientes cinco tipos de candidatos en orden:

- 1) Predictor del vector de movimiento espacial (MVP) de las CU espaciales vecinas;
- 2) MVP temporal de CU alocadas;
- 3) El MVP en función del historial de una tabla de primero en entrada, primero en salida (FIFO);
- 4) MVP promedio en pares; y
- 5) Vectores de movimiento cero (MV).

[0087] El tamaño de la lista de fusión se señala en una cabecera de conjunto de parámetros de secuencia (SPS). En un ejemplo, el tamaño máximo permitido de la lista de fusión es de 6 candidatos. Para cada CU codificada en modo de fusión, se codifica un índice de candidatos de fusión seleccionados mediante binarización unaria truncada (TU).

[0088] El proceso de derivación de cada categoría de candidatos de fusión se

proporciona en esta sección. Como se lleva a cabo en HEVC, VVC también soporta la derivación paralela de las listas de candidatos de fusión para todas las CU dentro de un cierto tamaño de área.

[0089] Derivación espacial de candidatos

[0090] La derivación de los candidatos de fusión espacial en VVC es la misma que en HEVC, con la excepción de que las posiciones de los dos primeros candidatos de fusión están intercambiadas. La Figura 2 ilustra posiciones de ejemplo de candidatos de fusión espacial. Se seleccionan un máximo de cuatro candidatos de fusión de los candidatos ubicados en las posiciones representadas en la Figura 2 en relación a la CU 400 actual. El orden de derivación es: B_0 , A_0 , B_1 , A_1 y B_2 . La posición B_2 se considera solo cuando una o más CU en las posiciones B_0 , A_0 , B_1 , A_1 no están disponibles (por ejemplo, porque la CU pertenece a otro segmento o mosaico) o cuando se intracodifica la CU. Después de añadir un candidato de fusión en la posición A_1 , la adición de los restantes candidatos de fusión se somete a una comprobación de redundancia que garantiza que los candidatos con la misma información de movimiento se excluyen de la lista para mejorar la eficacia de codificación.

[0091] Derivación temporal de candidatos

[0092] En esta técnica, solo se agrega un candidato a la lista. Particularmente, en la derivación de este candidato de fusión temporal, se deriva un vector de movimiento escalado en función de la CU coubicada que pertenece a la imagen de referencia colocada. La Figura 3 ilustra un ejemplo de escalado de vector de movimiento para un candidato de fusión temporal. La lista de imágenes de referencia que se utilizará para la derivación de la CU coubicada se señala explícitamente en la cabecera del corte. El vector de movimiento escalado 412 para el candidato de fusión temporal se obtiene como se ilustra en la línea punteada de la Figura 3 que se escala del vector de movimiento de la CU 416 coubicada mediante el uso de las distancias de recuento de orden de imagen (POC), t_b y t_d . El vector de movimiento 414 es el vector de movimiento entre la CU actual 400 y la CU coubicada 416. La distancia t_b se define como la diferencia de POC entre la imagen de referencia 404 de la imagen actual y la imagen actual 406. La distancia t_d se define como la diferencia de POC entre la imagen de referencia 402 de la imagen coubicada y la imagen coubicada 408. El índice de imagen de referencia del candidato de fusión se establece igual a cero.

[0093] La Figura 4 ilustra posiciones de candidatos de ejemplo para los candidatos de fusión temporal. La puede seleccionarse una posición para el candidato temporal para

la CU actual 400 se selecciona entre los candidatos C_0 y C_1 , como se representa en la Figura 4. Si la CU en la posición C_0 no está disponible (por ejemplo, está intracodificada, o está fuera de la fila actual de CTU), se utiliza la posición C_1 . De lo contrario, la posición C_0 se utiliza en la derivación del candidato de fusión temporal.

[0094] Derivación de candidatos de fusión en función del historial

[0095] Los candidatos de fusión de MVP en función del historial (HMVP) se agregan a la lista de fusión después de los candidatos de fusión espacial (por ejemplo, MVP espacial) y el candidato de fusión temporal (por ejemplo, MVP temporal o TMVP). En HMVP, la información de movimiento de un bloque previamente codificado se almacena en una tabla y se utiliza como MVP para la CU actual. La tabla con múltiples candidatos HMVP se mantiene durante el proceso de codificación/decodificación. La tabla se reinicia (por ejemplo, se vacía) cuando se encuentra una nueva fila de CTU. Cada vez que existe una CU intercodificada no sub-bloque, la información de movimiento asociada se agrega a la última entrada de la tabla como un nuevo candidato HMVP.

[0096] El tamaño de tabla HMVP S se establece como 6, que indica que se pueden agregar hasta 6 candidatos HMVP a la tabla. Cuando se inserta un nuevo candidato de movimiento a la tabla, se utiliza una regla de primero en entrada y primero en salida (FIFO) limitada, en donde primero se aplica una verificación de redundancia para encontrar si existe un HMVP idéntico en la tabla. Si se encuentra, el HMVP idéntico se elimina de la tabla y todos los candidatos de HMVP posteriores se mueven hacia adelante.

[0097] Los candidatos HMVP se pueden usar en el proceso de construcción de la lista de candidatos de fusión. Los últimos candidatos de HMVP de la tabla se comprueban en orden e insertarse en la lista de candidatos después del candidato de TMVP. Se aplica una verificación de redundancia en los candidatos HMVP para comparar la información de movimiento del candidato HMVP con el candidato de fusión espacial o temporal.

[0098] Para reducir la cantidad de operaciones de verificación de redundancia, se pueden introducir las siguientes simplificaciones:

- a. El número de candidatos de HMVP utilizados para la generación de listas de fusión se establece como $(N \leq 4) ? M : (8 - N)$, en donde N indica el número de candidatos existentes en la lista de fusión y M indica el número de candidatos de HMVP disponibles en la tabla.
- b. Una vez que el número total de candidatos de fusión disponibles

alcanza el número máximo permitido de candidatos de fusión menos 1, el proceso de construcción de la lista de candidatos de fusión a partir de HMVP se da por terminado.

[0099] Derivación de candidatos de fusión promedio en pares

[0100] Los candidatos promedio en pares se generan al promediar los pares predefinidos de candidatos en la lista de candidatos de fusión existente. En un ejemplo, los pares predefinidos se definen como $\{(0, 1), (0, 2), (1, 2), (0, 3), (1, 3), (2, 3)\}$, donde los números denotan los índices de fusión a la lista de candidatos de fusión. Los vectores de movimiento promediados se calculan por separado para cada lista de referencias. Si ambos vectores de movimiento están disponibles en una lista, los dos vectores de movimiento se promedian incluso cuando ambos apuntan a diferentes imágenes de referencia; si solo un vector de movimiento está disponible, se usa uno directamente; si no hay un vector de movimiento disponible, se establece la lista como inválida.

[0101] Cuando la lista de fusión no está completa después de agregar los candidatos de fusión promedio en pares, se insertan los MVP al final de la lista hasta alcanzar el número máximo de candidatos de fusión.

[0102] Bipredicción con ponderación a nivel de CU (BCW)

[0103] En HEVC, la señal de bipredicción se genera al promediar dos señales de predicción obtenidas de dos imágenes de referencia diferentes y/o usar dos vectores de movimiento diferentes. En VVC, el modo de bipredicción se extiende más allá del promedio simple para permitir el promedio ponderado de las dos señales de predicción.

$$P_{bi-pred} = ((8 - w) * P_0 + w * P_1 + 4) \gg 3$$

[0104] Se permiten cinco ponderaciones en la bipredicción de promedio ponderado, $w \in \{-2, 3, 4, 5, 10\}$. Para cada CU bipredicha, la ponderación w se determina de una de dos maneras: 1) para una CU que no es de fusión, se señala el índice de ponderación tras después de la diferencia del vector de movimiento; 2) para una CU de fusión, el índice de ponderación se infiere a partir de bloques vecinos, en función del índice de candidatos de fusión. Solo se aplica BCW a las CU con 256 o más muestras de luminancia (por ejemplo, el ancho de CU por la altura de CU es mayor o igual de 256). Para imágenes de bajo retraso, se utilizan las 5 ponderaciones. Para imágenes sin bajo retraso, se utilizan 3 ponderaciones ($w \in \{3, 4, 5\}$)

– En el codificador, se aplican algoritmos de búsqueda rápida para encontrar el

índice de ponderación sin aumentar significativamente la complejidad del codificador. Estos algoritmos se sintetizan de la siguiente manera. Se pueden obtener detalles adicionales a partir del software de modelo de prueba VVC (VTM) y en Yu-Chi Su, et al. "CE4-related: Generalized bi-prediction improvements combined from JVET-L0197 and JVET-L0296," Joint Video Experts Team (JVET) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 12.^a reunión: Macao, CN, 3-12 de octubre de 2018 (en adelante, "JVET-L0646"). Cuando se combina con AMVR, las ponderaciones desiguales sólo se comprueban condicionalmente para las precisiones de vectores de movimiento de 1 y 4 píxeles si la imagen actual es de bajo retraso.

- Cuando se combina con la estimación de movimiento afín (ME), la ME afín se realizará para ponderaciones desiguales si y sólo si el modo afín se selecciona como el mejor modo actual.
- Cuando las dos imágenes de referencia en bipredicción son iguales, las ponderaciones desiguales solo se comprueban condicionalmente.
- No se buscan ponderaciones desiguales cuando se cumplen determinadas condiciones (en función de la POC entre la imagen actual y sus imágenes de referencia, el QP de codificación y el nivel temporal).

[0105] El índice de ponderación BCW se codifica mediante el uso de una ubicación codificada por contexto seguida de ubicaciones codificadas por derivación. La primera ubicación codificada por contexto indica si se utilizan ponderaciones iguales y si se utiliza una ponderación desigual, se señalizan ubicaciones adicionales mediante el uso de la codificación de derivación para indicar qué ponderación desigual se utiliza.

[0106] La predicción ponderada (WP) es una herramienta de codificación admitida por las normas H.264/AVC y HEVC para codificar eficazmente contenidos de video con desvanecimiento. También se ha añadido soporte para WP en la norma de VVC. La WP permite que los parámetros de ponderación (ponderación e inclinación) se señalicen para cada imagen de referencia en cada una de las listas de imágenes de referencia L0 y L1. Luego, durante la compensación del movimiento, se aplican la ponderación y la inclinación de la imagen de referencia correspondientes. La WP y la BCW se designan para distintos tipos de contenidos de video. Para evitar interacciones entre la WP y la BCW, que complicarían el diseño del decodificador VVC, si una CU utiliza WP, entonces el índice de ponderación de BCW no se señaliza, y se infiere que w es 4 (por ejemplo, se aplica la ponderación igual). Para una CU de fusión, el índice de ponderación se infiere a partir de bloques vecinos, en función del

índice de candidatos de fusión. Esto se puede aplicar tanto en el modo de fusión normal como en el modo de fusión afín heredado. Para el modo de fusión afín construido, la información de movimiento afín se construye en función de la información de movimiento de hasta 3 bloques. El índice de BCW para una CU que utilice el modo de fusión afín construido simplemente se establece igual al índice de BCW del primer punto de control de MV.

[0107] En VVC, CIIP y BCW no pueden aplicarse conjuntamente para una CU. Cuando se codifica una CU con el modo CIIP, el índice de BCW de la CU actual se establece en 2, por ejemplo, igual peso.

[0108] Refinamiento del vector de movimiento lateral del decodificador en VVC

[0109] En VVC, puede aplicarse el refinamiento del vector de movimiento lateral del decodificador (DMVR) basado en BM para aumentar la precisión de los MV de un candidato a fusión de bipredicción. El método BM calcula la suma de diferencias absolutas (SAD) entre los dos bloques candidatos de la lista de imágenes de referencia L0 y la lista L1. LA FIGURA 5 ilustra un ejemplo de comparación bilateral. Como se ilustra en la Figura 5, se calcula la SAD entre los bloques 500 y 502 en función de cada candidato de MV alrededor del MV inicial. El candidato de MV con la SAD más baja se convierte en el MV refinado y se utiliza para generar la señal bipredicha. A la SAD de los MV iniciales se le resta $\frac{1}{4}$ del valor de la SAD para que sirva como término de regularización. Las distancias temporales (por ejemplo, la diferencia de orden de imágenes (POC)) de dos imágenes de referencia a la imagen actual deben ser las mismas, por lo tanto, el MVD0 es justo el signo opuesto del MVD1.

[0110] El rango de búsqueda de refinamiento es dos muestras de luminancia de número entero del MV inicial. La búsqueda incluye la etapa de búsqueda de inclinación de la muestra de número entero y la etapa de refinamiento de la muestra fraccional. Se aplica la búsqueda completa de 25 puntos para la búsqueda de desviación de muestras enteras. Primero se calcula la SAD del par de MV iniciales. Si la SAD del par de MV iniciales es menor que un umbral, finaliza la etapa de la muestra de número entero de DMVR. En cambio, las SAD de los 24 puntos restantes se calculan y se verifican en el orden de exploración de trama. El punto con la SAD más pequeña se selecciona como la salida de la etapa de búsqueda de inclinación de la muestra de número entero.

[0111] A la búsqueda de muestras enteras le sigue el refinamiento de muestras fraccionarias. Para reducir la complejidad del cálculo, el refinamiento fraccional de la

muestra se obtiene mediante la ecuación paramétrica de la superficie de error, en lugar de utilizar una búsqueda adicional con una comparación SAD. El refinamiento de la muestra fraccional se invoca condicionalmente en función de la salida de la etapa de búsqueda de la muestra de número entero. Cuando finaliza la etapa de búsqueda de la muestra de número entero con el centro que tiene la SAD más pequeña en la primera o la segunda búsqueda de iteración, se aplica adicionalmente el refinamiento de la muestra fraccional.

[0112] En la estimación de desplazamientos de subpíxeles basada en la superficie de error paramétrica, el costo de la posición central y los costos en cuatro posiciones vecinas desde el centro se utilizan para ajustar una ecuación de superficie de error parabólica bidimensional de la siguiente forma:

$$E(x, y) = A(x - x_{min})^2 + B(y - y_{min})^2 + C, \quad (1)$$

donde (x_{min}, y_{min}) corresponde a la posición fraccionaria con menor costo y C corresponde al valor de costo mínimo. Al resolver las ecuaciones anteriores mediante el uso del valor del costo de los cinco puntos de búsqueda, el (x_{min}, y_{min}) se calcula como:

$$x_{min} = (E(-1,0) - E(1,0)) / (2(E(-1,0) + E(1,0) - 2E(0,0))) \quad (2)$$

$$y_{min} = (E(0,-1) - E(0,1)) / (2(E(0,-1) + E(0,1) - 2E(0,0))) \quad (3)$$

[0113] Los valores de x_{min} y y_{min} están limitados a estar entre - 8 y 8 ya que todos los valores de costo son positivos y el valor más pequeño es $E(0,0)$. Esto corresponde a un desplazamiento de medio píxel con una precisión de 1/16 de píxel en VVC. Los valores fraccionarios (x_{min}, y_{min}) calculados se añaden a la VM de refinamiento de distancia entera para obtener el MV delta de refinamiento con precisión de subpíxel.

[0114] En VVC, la resolución de los MV es de 1/16 muestras de luminancia. Las muestras en la posición fraccional se interpolan mediante el uso de un filtro de interpolación de 8 tomas. En DMVR, los puntos de búsqueda rodean el MV inicial de píxel fraccional con la inclinación de la muestra de número entero. Por lo tanto, las muestras de esas posiciones fraccionarias se interpolan para el proceso de búsqueda de DMVR. Para reducir la complejidad del cálculo, se utiliza el filtro de interpolación bilineal para generar las muestras fraccionarias para el proceso de búsqueda en DMVR. Otro efecto es que, al utilizar un filtro bilineal con un rango de búsqueda de 2 muestras, el DVMR no accede a más muestras de referencia en comparación con el

proceso normal de compensación de movimiento. Una vez obtenido el MV refinado con el proceso de búsqueda DMVR, se aplica el filtro de interpolación normal de 8 tomas para generar la predicción final. Para no acceder a más muestras de referencia en relación con un proceso normal de compensación de movimiento, las muestras que no son necesarias para el proceso de interpolación en función del MV original, pero sí para el proceso de interpolación en función del MV refinado, se rellenarán a partir de esas muestras disponibles.

[0115] Cuando la anchura y/o la altura de una CU son superiores a 16 muestras de luminancia, la CU se dividirá a su vez en subbloques con anchura y/o altura iguales a 16 muestras de luminancia para el proceso DMVR.

[0116] En VVC, DMVR puede aplicarse a las CU codificadas con los siguientes modos y características:

- Modo de fusión a nivel de CU con MV de bipredicción
- Una imagen de referencia está en el pasado y otra imagen de referencia está en el futuro con respecto a la imagen actual
- Las distancias (por ejemplo, diferencia de POC) entre dos imágenes de referencia y la imagen actual son iguales
- Ambas imágenes de referencia son imágenes de referencia a corto plazo
- La CU tiene más de 64 muestras de luminancia
- Tanto la altura como la anchura de la CU son mayores o iguales que 8 muestras de luminancia
- El índice de peso BCW indica igual peso
- WP no está activado para el bloque actual
- El modo CIIP no se utiliza para el bloque actual

[0117] Flujo óptico bidireccional

[0118] El flujo óptico bidireccional (BDOF) se utiliza para refinar la señal de bipredicción de muestras de luminancia en una CU a nivel de subbloque 4×4. Como su nombre lo indica, el modo de BDOF se basa en el concepto de flujo óptico, que supone que el movimiento de un objeto es fluido. Para cada subbloque de 4×4, se calcula un refinamiento de movimiento (v_x, v_y) al minimizar la diferencia entre las muestras de predicción de L0 y L1. Luego se utiliza el refinamiento de movimiento para ajustar los valores de muestra bipredichos en el subbloque de 4×4. Las etapas siguientes se aplican en el proceso de BDOF.

[0119] Primero, se computan los gradientes horizontales y verticales, $\frac{\partial I^{(k)}}{\partial x}(i, j)$ y $\frac{\partial I^{(k)}}{\partial y}(i, j)$, $k = 0, 1$, de las dos señales de predicción al calcular directamente la diferencia entre las dos muestras cercanas, por ejemplo,

$$\begin{aligned}\frac{\partial I^{(k)}}{\partial x}(i, j) &= (I^{(k)}(i+1, j) \gg \text{shift1}) - (I^{(k)}(i-1, j) \gg \text{shift1}) \\ \frac{\partial I^{(k)}}{\partial y}(i, j) &= (I^{(k)}(i, j+1) \gg \text{shift1}) - (I^{(k)}(i, j-1) \gg \text{shift1}),\end{aligned}\tag{1-6-1}$$

cuando $I^{(k)}(i, j)$ están los valores de muestra en la coordenada (i, j) de la señal de predicción en la lista k , $k = 0, 1$, y shift1 se calcula en función de la profundidad de bits de luminancia, bitDepth, y shift1 se configura para ser igual a 6.

[0120] Luego, se calcula la autocorrelación y la correlación cruzada de los gradientes, S_1 , S_2 , S_3 , S_5 y S_6 como:

$$\begin{aligned}S_1 &= \sum_{(i,j) \in \Omega} |\psi_x(i, j)|, \quad S_3 = \sum_{(i,j) \in \Omega} \theta(i, j) \cdot (-\text{sign}(\psi_x(i, j))) \\ S_2 &= \sum_{(i,j) \in \Omega} \psi_x(i, j) \cdot \text{sign}(\psi_y(i, j)) \\ S_5 &= \sum_{(i,j) \in \Omega} |\psi_y(i, j)|, \quad S_6 = \sum_{(i,j) \in \Omega} \theta(i, j) \cdot (-\text{sign}(\psi_y(i, j))),\end{aligned}\tag{1-6-2}$$

donde

$$\begin{aligned}\psi_x(i, j) &= \left(\frac{\partial I^{(1)}}{\partial x}(i, j) + \frac{\partial I^{(0)}}{\partial x}(i, j) \right) \gg \text{shift3} \\ \psi_y(i, j) &= \left(\frac{\partial I^{(1)}}{\partial y}(i, j) + \frac{\partial I^{(0)}}{\partial y}(i, j) \right) \gg \text{shift3} \\ \theta(i, j) &= (I^{(0)}(i, j) \gg \text{shift2}) - (I^{(1)}(i, j) \gg \text{shift2}),\end{aligned}\tag{1-6-3}$$

donde Ω es una ventana de 6×6 alrededor del subbloque de 4×4, el valor de shift2 se configura para ser igual a 4, y el valor de shift3 se configura para ser igual a 1.

[0121] Luego se deriva el refinamiento de movimiento (v_x, v_y) mediante el uso de los términos de correlación cruzada y autocorrelación mediante las siguientes ecuaciones:

$$v_x = S_1 > 0 ? \text{clip3}(-th'_{BIO}, th'_{BIO}, -((S_3 \ll 2) \gg \lfloor \log_2 S_1 \rfloor)) : 0\tag{1-6-4}$$

$$v_y = S_5 > 0? \text{clip3}\left(-th'_{BIO}, th'_{BIO}, -\left(\left((S_6 \ll 2) - ((v_x \cdot S_2) \gg 1)\right) \gg \lfloor \log_2 S_5 \rfloor\right)\right): 0,$$

donde, $th'_{BIO} = 1 \ll 4$. $\lfloor \cdot \rfloor$ es la función piso.

[0122] En función del refinamiento de movimiento y los gradientes, se calcula el siguiente ajuste para cada muestra en el subbloque de 4×4:

$$b(x, y) = v_x \cdot \left(\frac{\partial I^{(1)}(x, y)}{\partial x} - \frac{\partial I^{(0)}(x, y)}{\partial x} \right) + v_y \cdot \left(\frac{\partial I^{(1)}(x, y)}{\partial y} - \frac{\partial I^{(0)}(x, y)}{\partial y} \right) \quad (1-6-5)$$

[0123] Finalmente, las muestras de BDOF de la CU se calculan mediante el ajuste de las muestras de bipredicción de la siguiente manera:

$$pred_{BDOF}(x, y) = (I^{(0)}(x, y) + I^{(1)}(x, y) + b(x, y) + o_{offset}) \gg shift5, \quad (1-6-6)$$

donde se configura shift5 igual a Max(3, 15 - BitDepth) y la variable o_{offset} se configura igual a (1 << (shift5- 1)).

[0124] Estos valores se seleccionan de manera tal que los múltiplos en el proceso de BDOF no superan 15 bits, y el ancho de bits máximo de los parámetros intermedios en el proceso de BDOF se mantiene dentro de 32 bits.

[0125] Para derivar los valores de gradiente, es posible que se necesiten generar algunas muestras de predicción $I^{(k)}(i, j)$ en la lista k ($k = 0, 1$) fuera de los límites

actuales de la CU. LA FIGURA 6 ilustra un ejemplo de una región de unidad de codificación extendida utilizada en el flujo óptico bidireccional. De acuerdo con lo mostrado en la Figura 6, BDOF usa una fila/columna extendida alrededor de los límites de CU 600. Para controlar la complejidad computacional de la generación de las muestras de predicción fuera de los límites, las muestras de predicción en el área extendida (posiciones blancas) se generan tomando las muestras de referencia en las posiciones enteras cercanas (mediante la operación de psio() en las coordenadas) directamente sin interpolación, y el filtro normal de interpolación de compensación de movimiento de 8 tomas se utiliza para generar muestras de predicción dentro de la CU (posiciones grises). Estos valores de muestra extendidos se utilizan solamente en un cálculo de gradiente. Para las etapas restantes en el proceso de BDOF, si se necesitan muestras y valores de gradiente fuera de los límites de la CU, se pueden rellenar (por ejemplo, repetidos) a partir de los vecinos más cercanos.

[0126] El BDOF se utiliza para refinar la señal de bipredicción de una CU en el nivel de subbloque 4×4. Se puede aplicar BDOF a una CU si cumple con todas las siguientes condiciones:

- La CU se codifica mediante el uso del modo de bipredicción "verdadero", por ejemplo, una de las dos imágenes de referencia es anterior a la imagen actual en el orden de visualización y la otra es posterior a la imagen actual en el orden de visualización

- La CU no se codifica mediante el uso del modo afín o el modo de fusión SbTMVP

- La CU tiene más de 64 muestras de luminancia

- Tanto la altura como la anchura de la CU son mayores o iguales que 8 muestras de luminancia

- El índice de peso BCW indica igual peso

- WP no está activado para la CU actual

- El modo CIIP no se utiliza para la CU actual

[0127] El refinamiento de movimiento lateral del decodificador de pase múltiple

[0128] En la reunión de JVET-V se estableció un modelo de compresión mejorado (ECM, https://vcgit.hhi.fraunhofer.de/ecm/VVCSoftware_VTM/-/tree/ECM) para estudiar técnicas de compresión más allá del VVC. En ECM, se adoptó una técnica de refinamiento de movimiento lateral del decodificador de pase múltiple, que se describe en Yao-Jen Chang, et al. "Eficacia de la compresión más allá del VVC," Joint Video Experts Team (JVET) de ITU-T SG 16 WP 3 y ISO/IEC JTC 1/SC 29/WG 11, 21ª Reunión, por teleconferencia, 6-15 de enero de 2021 (en adelante, "JVET_U0100") para reemplazar el DMVR en VVC. En el primer pase, el BM (similar a la de DMVR en VVC) se aplica al bloque de codificación. En el segundo pase, el BM se aplica a cada subbloque de 16x16 dentro del bloque de codificación con los MV refinados en el primer pase como MV iniciales. En el tercer pase, un MV en cada subbloque de 8x8 pueden refinarse aún más al aplicar flujo óptico bidireccional (BDOF).

[0129] Primer pase - Comparación bilateral basado en bloques Refinamiento de MV

[0130] En el primer pase, se deriva un MV refinado al aplicar BM a un bloque de codificación. De forma similar al refinamiento del vector de movimiento del lado del decodificador (DMVR), en la operación de bipredicción, se busca un MV refinado en torno a dos MV iniciales (por ejemplo, MV0 y MV1) en las listas de imágenes de referencia L0 y L1. Los MV refinados (MV0_pass1 y MV1_pass1) se derivan en torno a los MV iniciales en función del costo mínimo de comparación bilateral entre los dos bloques de referencia en L0 y L1.

[0131] El BM incluye la realización de una búsqueda local para derivar la precisión de

la muestra entera de intDeltaMV. La búsqueda local aplica un patrón de búsqueda cuadrado 3×3 para recorrer en bucle el intervalo de búsqueda [-sHor, sHor] en dirección horizontal y [-sVer, sVer] en dirección vertical, en donde los valores de sHor y sVer vienen determinados por la dimensión del bloque, y el valor máximo de sHor y sVer es 8.

[0132] El costo de comparación bilateral se calcula como: $\text{bilCost} = \text{mvDistanceCost} + \text{sadCost}$. Cuando el tamaño de bloque $\text{cbW} * \text{cbH}$ es superior a 64, se aplica una función de costo SAD de eliminación media (MRSAD) para eliminar el efecto DC de la distorsión entre bloques de referencia. Cuando el bilCost en el punto central del patrón de búsqueda 3×3 tiene el costo mínimo, se termina la búsqueda local intDeltaMV. En caso contrario, el punto de búsqueda de costo mínimo actual se convierte en el nuevo punto central del patrón de búsqueda 3×3 y el codificador de video sigue buscando el costo mínimo hasta que se alcanza el final del intervalo de búsqueda.¹¹

[0133] El refinamiento fraccional de la muestra existente se aplica además para obtener el deltaMV final. Los MV refinados después del primer pase pueden derivarse entonces como:

- $\text{MV0_pass1} = \text{MV0} + \text{deltaMV}$
- $\text{MV1_pass1} = \text{MV1} - \text{deltaMV}$

[0134] Segundo pase - Comparación bilateral basado en bloques Refinamiento de MV

[0135] En el segundo pase, se deriva un MV refinado al aplicar BM a un subbloque de cuadrícula de 16×16. Para cada subbloque, se busca un MV refinado alrededor de los dos MV (MV0_pass1 y MV1_pass1), obtenidos en el primer pase, en la lista de imágenes de referencia L0 y L1. Los MV refinados ($\text{MV0_pass2}(\text{sblIdx2})$ y $\text{MV1_pass2}(\text{sblIdx2})$), en donde $\text{sblIdx2} = 0, \dots, N-1$, es el índice para el subbloque, se derivan en función del mínimo costo de comparación bilateral entre los dos subbloques de referencia en L0 y L1.

[0136] Para cada subbloque, BM incluye la realización de una búsqueda completa para derivar la precisión de la muestra entera intDeltaMV. La búsqueda completa tiene un rango de búsqueda [-sHor, sHor] en dirección horizontal y [-sVer, sVer] en dirección vertical, en donde los valores de sHor y sVer vienen determinados por la dimensión del bloque, y el valor máximo de sHor y sVer es 8.

[0137] El costo de la comparación bilateral se calcula mediante la aplicación de un factor de costo a la suma de las diferencias transformadas absolutas (SATD) entre dos subbloques de referencia: $\text{bilCost} = \text{satdCost} * \text{costFactor}$. LA FIGURA 7 ilustra un

ejemplo de región de búsqueda de diamante 700 para el refinamiento de movimiento lateral del decodificador multipase. El área de búsqueda $(2 \cdot s_{Hor} + 1) \cdot (2 \cdot s_{Ver} + 1)$ se divide en 5 regiones de búsqueda en forma de diamante, como se muestra en la Figura 7. A cada región de búsqueda se le asigna un valor de `costFactor`, que se determina por la distancia (`intDeltaMV`) entre cada punto de búsqueda y el MV inicial. Cada región diamante se procesa en un orden que comienza desde el centro del área de búsqueda. Dentro de cada región, los puntos de búsqueda se procesan en el orden de escaneado de trama, que comienza por la esquina superior izquierda hasta la esquina inferior derecha de la región. Cuando el `bilCost` mínimo dentro de la región de búsqueda actual es menor que un umbral igual a $sbW \cdot sbH$, se termina la búsqueda completa int-pel. En caso contrario, la búsqueda completa int-pel continúa en la siguiente región de búsqueda hasta que se examinan todos los puntos de búsqueda.

[0138] El refinamiento fraccional de muestras VVC DMVR existente se aplica además para derivar el `deltaMV(sbIdx2)` final. El MV refinado en el segundo pase se deriva entonces como:

- $MV0_pass2(sbIdx2) = MV0_pass1 + \text{deltaMV}(sbIdx2)$
- $MV1_pass2(sbIdx2) = MV1_pass1 - \text{deltaMV}(sbIdx2)$

[0139] Tercer pase - Refinamiento de MV de flujo óptico bidireccional basado en subbloques

[0140] En el tercer pase, se obtiene un MV refinado aplicando BDOF a un subbloque de cuadrícula de 8×8 . Para cada subbloque de 8×8 , se aplica el refinamiento BDOF para derivar los valores V_x y V_y escalados, sin recorte, partiendo del VM refinado del subbloque padre del segundo pase. La $\text{bioMv}(V_x, V_y)$ derivada se redondea con una precisión de $1/16$ de muestra y se recorta entre -32 y 32.

[0141] Los MV refinados ($MV0pass3(sbIdx3)$ y $MV1pass3(sbIdx3)$) en el tercer pase pueden derivarse como:

- $MV0_pass3(sbIdx3) = MV0_pass2(sbIdx2) + \text{bioMv}$
- $MV1_pass3(sbIdx3) = MV0_pass2(sbIdx2) - \text{bioMv}$

Se pueden encontrar más detalles de DMVR multipase en la solicitud de patente provisional de EE. UU. 63/129,221, presentada el 22 de diciembre de 2020.

[0142] Comparación de plantilla

[0143] La comparación de plantilla (TM) es un método de derivación de MV del lado del decodificador que refina la información de movimiento de la CU actual encontrando la coincidencia más cercana entre una plantilla (por ejemplo, bloques vecinos superiores y/o izquierdos de la CU actual) en la imagen actual y un bloque (por

ejemplo, del mismo tamaño que la plantilla) en una imagen de referencia. LA FIGURA 8 ilustra un ejemplo de comparación de plantillas llevado a cabo en un área de búsqueda alrededor de un vector de movimiento inicial. Como se ilustra en la Figura 8, se buscará un MV mejor alrededor del movimiento inicial 800 de la CU actual 802 dentro de un rango de búsqueda de $[-8, +8]$ -pel 806 en el marco de referencia 808. Con un candidato de AMVP seleccionado en función del error de comparación inicial, el MVP de AMVP se refina mediante comparación de plantillas. Con un candidato a fusión indicado por el índice de fusión señalado, sus MV fusionados correspondientes a L0 y L1 se refinan de forma independiente mediante la comparación de plantillas y, a continuación, el vector de movimiento menos preciso se refina de nuevo con el vector de movimiento mejor como vector de movimiento inicial (por ejemplo, un vector de movimiento anterior).

[0144] Función de costo: Cuando un vector de movimiento apunta a una posición de muestra fraccionaria, puede utilizarse la interpolación compensada por movimiento. Para reducir la complejidad, se utiliza la interpolación bilineal, en lugar de la interpolación DCT-IF de 8 pasos, para generar plantillas en imágenes de referencia. El costo de comparación C de la comparación de plantillas se puede calcular de la siguiente manera:

$$C = SAD + w * (|MV_x - MV^s_x| + |MV_y - MV^s_y|),$$

donde w es un factor de ponderación que puede establecerse en un número entero tal como 0, 1, 2, 3 o 4, y MV y MV^s indican el MV probado actualmente y el MV inicial (por ejemplo, un candidato a MVP en el modo AMVP o un movimiento fusionado en el modo fusión), respectivamente. La SAD se utiliza como costo de comparación de la comparación de plantillas.

[0145] Cuando se usa TM, se refina la información del movimiento solamente mediante el uso de las muestras de luminancia. La información de movimiento derivada se utilizará tanto para luminancia como para crominancia para la interpredicción con compensación de movimiento. Una vez decidido el MV, se realiza la compensación final del movimiento mediante el uso de un filtro de interpolación de 8 tomas para los componentes de luminancia y un filtro de interpolación de 4 tomas para los componentes de crominancia.

[0146] Método de búsqueda: El refinamiento de MV es una búsqueda de MV basada en patrones con el criterio del costo de comparación de plantillas y una estructura jerárquica. Se admiten dos patrones de búsqueda para el refinamiento de MV – una búsqueda de diamante y una búsqueda de cruz. La estructura jerárquica especifica un

proceso iterativo para refinar el MV, comenzando con una precisión MVD gruesa (por ejemplo, un cuarto de píldora) y terminando con una precisión MVD fina (por ejemplo, 1/8 de píldora). El MV se busca directamente con una precisión MVD de un cuarto de muestra de luminancia con un patrón de diamante, seguida de una precisión MVD de un cuarto de muestra de luminancia con un patrón de cruz. A esto le sigue un refinamiento MVD de un octavo de muestra de luminancia con un patrón cruzado. El rango de búsqueda del refinamiento de MV puede configurarse para ser igual (-8, +8) a las muestras de luminancia alrededor del MV inicial. Cuando el bloque actual se codifica mediante el uso de la bipredicción, ambos MV se refinan de forma independiente y, a continuación, el mejor de estos (en términos de costo de comparación) se establece como MV previo para refinar aún más el otro MV con valores de peso BCW.

[0147] En concreto, la comparación de plantillas propuesta en Y-W Chen, et. al "Description of SDR, HDR and 360° video coding technology proposal by Qualcomm and Technicolor - low and high complexity versions," Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 y ISO/IEC JTC 1/SC 29/WG 11, Reunión 10°: San Diego, Estados Unidos, 10-20 de abril de 2018 (en adelante, "JVET-J0021") y JVET-U0100 se aplica al modo AMVP y al modo de fusión en algunas propuestas de ejemplo.

[0148] La lista de candidatos a la fusión regular se construye siguiendo el orden de los siguientes candidatos:

- MVP espaciales de CU vecinas espaciales: se seleccionan los 4 primeros candidatos disponibles en el orden de CU vecina superior (A), CU vecina izquierda (L), CU vecina superior derecha (AR), CU vecina inferior izquierda (LB), CU vecina superior izquierda (LA).
- MVP temporal de CU colocadas; Solo se agrega un candidato.
- MVP en función del historial: La información de movimiento de un bloque codificado previamente se almacena en una tabla y se utiliza como MVP para la CU actual.
- Promedio por pares de MVP: Los candidatos promedio por pares se generan promediando pares de candidatos predefinidos en la lista de candidatos a la fusión regular existente.
- MV cero.

[0149] Cuando se aplica la TM a bloques codificados mediante el modo de fusión, se construye una lista separada de candidatos a la fusión TM refinando los MVP de los candidatos a la fusión regulares en función de la comparación de plantillas.

[0150] En el proceso de búsqueda de movimiento del lado del decodificador, ya sea para DMVR basado en comparación bilateral o en comparación de plantillas, se puede definir un criterio de costo y utilizarlo para calcular los costos de las posiciones candidatas dentro de un área de búsqueda predefinida. La información de movimiento asociada a la posición candidata que tiene el costo mínimo es el resultado del proceso de búsqueda iterativa. El proceso de búsqueda se puede conducir en múltiples iteraciones. En cada iteración, se evalúa un subconjunto de posiciones de candidatos dentro de un área de búsqueda. Pueden aplicarse determinados patrones para seleccionar las posiciones candidatas de cada subconjunto. Por ejemplo, la búsqueda cuadrada, la búsqueda diamante y la región diamante de la Figura 7. Generalmente, la distancia entre la posición inicial y una posición de candidato aumenta tras cada iteración. En otras palabras, las posiciones de candidatos en la iteración de búsqueda actual pueden estar más alejada del punto inicial que la posición de la iteración anterior.

[0151] Se puede definir una variable minCost para controlar el costo mínimo. La variable minCost puede representar el mínimo calculado durante el proceso de búsqueda iterativa. Esta divulgación puede referirse a minCost como el costo mínimo del proceso de búsqueda iterativa determinado hasta el momento. Se pueden aplicar algunos métodos de terminación anticipada para poner fin a un proceso de búsqueda. Por ejemplo, el proceso se termina si el minCost es menor que un umbral, o el proceso se termina si la mejor posición candidata con el costo mínimo no ha cambiado de un subconjunto al siguiente (por ejemplo, de una iteración a la siguiente).

[0152] Esta divulgación describe técnicas para la terminación anticipada de un proceso de búsqueda de técnicas de derivación de vectores de movimiento del lado del decodificador para reducir la complejidad computacional del proceso de búsqueda. Las técnicas de esta divulgación pueden utilizarse con cualquiera de las técnicas de derivación de vectores de movimiento del lado del decodificador que impliquen un proceso de búsqueda. En un ejemplo, las técnicas de la presente divulgación pueden aplicarse tanto en la comparación bilateral de DMVR como en la comparación de plantillas. En otros ejemplos, las técnicas de la presente divulgación también pueden utilizarse junto con otros métodos de terminación anticipada, por ejemplo, como se describió anteriormente.

[0153] En un ejemplo de la divulgación, el codificador de video 200 y el decodificador de video 300 pueden calcular el valor de una segunda variable minCostPrev para realizar un seguimiento del costo mínimo en una iteración de búsqueda anterior. En un

ejemplo, el número máximo de iteraciones deberá ser superior a 2 para que se aplique este ejemplo. En un ejemplo, el codificador de video 200 y el decodificador de video 300 pueden inicializar el valor de minCostPrev con un valor predeterminado, por ejemplo, cero.

[0154] El decodificador de video 300 puede determinar un costo mínimo a través de una iteración de búsqueda anterior y un costo mínimo global a través de una iteración de búsqueda actual. En algunos ejemplos, la iteración de búsqueda anterior puede preceder inmediatamente a la iteración de búsqueda actual. Por ejemplo, si la iteración de búsqueda actual es una 4ª iteración de búsqueda, la iteración de búsqueda anterior sería la 3ª iteración de búsqueda. En este ejemplo, un costo mínimo a través de una iteración de búsqueda anterior sería el costo mínimo a través de la 3ª iteración de búsqueda, mientras que un costo mínimo global a través de una iteración de búsqueda actual sería el costo mínimo a través de la 4ª iteración de búsqueda.

[0155] El codificador de video 200 puede determinar un costo mínimo a través de una iteración de búsqueda anterior y un costo mínimo global a través de una iteración de búsqueda actual. En algunos ejemplos, la iteración de búsqueda anterior puede preceder inmediatamente a la iteración de búsqueda actual. Por ejemplo, si la iteración de búsqueda actual es una 4ª iteración de búsqueda, la iteración de búsqueda anterior sería la 3ª iteración de búsqueda. En este ejemplo, un costo mínimo a través de la iteración de búsqueda anterior sería el costo mínimo a través de la 3ª iteración de búsqueda, mientras que un costo mínimo global a través de una iteración de búsqueda actual sería el costo mínimo a través de la 4ª iteración de búsqueda.

[0156] En otro ejemplo, se puede utilizar un costo mínimo de una iteración de búsqueda anterior en sí en lugar de un costo mínimo a través de la iteración de búsqueda anterior. Por ejemplo, si la iteración de búsqueda actual es una 4ª iteración de búsqueda, la iteración de búsqueda anterior sería la 3ª iteración de búsqueda. En este ejemplo, se puede utilizar un costo mínimo de la propia 3ª iteración de búsqueda en lugar del costo mínimo a través de la 3ª iteración de búsqueda.

[0157] En algunos ejemplos, para determinar un vector de movimiento para un bloque de los datos de video mediante un proceso de derivación de vector de movimiento del lado del decodificador, el decodificador de video 300 y/o el codificador de video 200 pueden configurarse para terminar un proceso de búsqueda iterativa anticipadamente de acuerdo con las técnicas analizadas en esta divulgación.

[0158] En las técnicas de terminación divulgadas, el codificador de video 200 y el decodificador de video 300 pueden estar configurados para considerar minCost y

minCostPrev para tomar una decisión sobre si se termina el proceso de búsqueda iterativa. En un ejemplo, el codificador de video 200 y el decodificador de video 300 están configurados para comparar la diferencia de $\text{minCostPrev} - \text{minCost}$ o $\text{abs}(\text{minCostPrev} - \text{minCost})$ con un umbral, y si la diferencia es menor que el umbral, el codificador de video 200 y el decodificador de video 300 terminan el proceso de búsqueda iterativa. Cuando el codificador de video 200 y el decodificador de video 300 deciden finalizar el proceso de búsqueda iterativa, el codificador de video 200 y el decodificador de video 300 pueden estar configurados para devolver el vector de movimiento que tenga el costo mínimo hasta ese punto del proceso de búsqueda.

[0159] En consecuencia, en un ejemplo, el codificador de video 200 y el decodificador de video 300 pueden estar configurados para terminar el proceso de búsqueda iterativa antes en función de una diferencia del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo global a través de la iteración de búsqueda actual en relación con un umbral de diferencia. Por ejemplo, si la iteración de búsqueda actual es la 4ª iteración de búsqueda y la iteración de búsqueda anterior es la 3ª iteración de búsqueda, y si una diferencia entre el costo mínimo a través de la 4ª iteración de búsqueda y el costo mínimo a través de la 3ª iteración de búsqueda está por debajo de un umbral de diferencia, se termina el proceso de búsqueda iterativa. En algunos ejemplos, el umbral de diferencia puede ser fijo o estar en función de un tamaño del bloque de datos de video. En algunos ejemplos, el umbral puede establecerse igual a S veces el área del bloque de datos de video, donde S es un factor de escala que puede predefinirse o señalarse en el flujo de bits.

[0160] En algunos ejemplos, el codificador de video 200 y el decodificador de video 300 pueden estar configurados para terminar el proceso de búsqueda iterativa anticipadamente en función de la comparación de un valor absoluto de la diferencia entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo global a través de la iteración de búsqueda actual con el umbral de diferencia.

[0161] En otros ejemplos, el codificador de video 200 y el decodificador de video 300 terminan el proceso de búsqueda iterativa anticipadamente cuando la diferencia entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo global a través de la iteración de búsqueda actual está por debajo de un umbral de diferencia puede reducir la complejidad computacional sin aumentar materialmente la distorsión.

[0162] En consecuencia, en un ejemplo de la divulgación, el decodificador de video 300 puede estar configurado para determinar un vector de movimiento para un bloque de datos de video mediante un proceso de derivación de vector de movimiento del

lado del decodificador que incluye un proceso de búsqueda iterativa, en el que el proceso de búsqueda iterativa incluye calcular un costo mínimo de una iteración de búsqueda anterior, y terminar el proceso de búsqueda iterativa anticipadamente en función del costo mínimo a través de la iteración de búsqueda anterior. El decodificador de video 300 puede entonces codificar el bloque de datos de video mediante el uso del vector de movimiento. En otro ejemplo, el decodificador de video 300 puede actualizar un costo mínimo determinado para el proceso de búsqueda iterativa. Para finalizar el proceso de búsqueda iterativa anticipadamente en función del costo mínimo de la iteración de búsqueda anterior, el decodificador de video 300 puede calcular además una diferencia o una relación entre el costo de la iteración de búsqueda anterior y el costo mínimo determinado para el proceso de búsqueda iterativa, y finalizar el proceso de búsqueda iterativa anticipadamente en función de la diferencia o la relación con respecto a un umbral.

[0163] En un ejemplo, las técnicas divulgadas pueden describirse de la siguiente manera. El codificador de video 200 y el decodificador de video 300 pueden inicializar minCostPrev y minCost, por ejemplo, para que tengan el mismo valor. En otro ejemplo, el valor de inicialización puede ser introducido desde el proceso anterior o fijado igual a un valor grande predefinido. Mientras el proceso de búsqueda iterativa no haya terminado y no se haya alcanzado el máximo de iteraciones, el codificador de video 200 y el decodificador de video 300 pueden realizar la siguiente iteración:

- 1) Establecer minCostPrev igual a minCost.
- 2) Evaluar el subconjunto actual de posiciones candidatas y determinar (por ejemplo, actualizar) el minCost.
- 3) Determina si debe finalizar en función de los valores de minCost y minCostPrev.

[0164] Obsérvese que también pueden aplicarse otras técnicas de terminación anticipada después de la etapa 2). Por ejemplo, pueden aplicarse otras técnicas de terminación anticipada antes de la etapa 3) o después.

[0165] En un ejemplo, la etapa 3) anterior incluye la evaluación de la diferencia entre minCost y minCostPrev. El codificador de video 200 y el decodificador de video 300 pueden decidir finalizar el proceso de búsqueda si $\text{minCostPrev} - \text{minCost}$ es inferior a un umbral.

[0166] En los ejemplos anteriores, el umbral puede ser un valor predefinido, fijo o señalado en el flujo de bits. En otros ejemplos, el umbral puede ser adaptativo en función del tamaño del bloque de entrada. Por ejemplo, el umbral puede establecerse

igual a S veces el área del bloque de entrada, en donde S es un factor de escala y puede predefinirse o señalarse en el flujo de bits.

[0167] En otro ejemplo, la diferencia de los dos costos utilizados anteriormente puede sustituirse por el coeficiente de costos. Por ejemplo, la etapa 3) se lleva a cabo evaluando la relación entre minCost y minCostPrev . El valor R es igual a $(\text{minCostPrev} - \text{minCost}) / \text{minCostPrev}$. El codificador de video 200 y el decodificador de video 300 pueden determinar finalizar el proceso de búsqueda si R es inferior a un umbral predefinido.

[0168] En consecuencia, en un ejemplo, el codificador de video 200 y el decodificador de video 300 pueden estar configurados para terminar el proceso de búsqueda iterativa anticipadamente puede terminar el proceso de búsqueda iterativa anticipadamente en función de una relación del costo mínimo a través de la iteración de búsqueda anterior al costo mínimo global a través de la iteración de búsqueda actual en relación con un umbral de relación. Por ejemplo, si la iteración de búsqueda actual es la 4ª iteración de búsqueda y la iteración de búsqueda anterior es la 3ª iteración de búsqueda, si la relación entre el costo mínimo a través de la 4ª iteración de búsqueda y el costo mínimo a través de la 3ª iteración de búsqueda está por debajo de un umbral de relación, el proceso de búsqueda iterativa finaliza. En algunos ejemplos, el umbral de relación puede ser fijo o estar en función de un tamaño del bloque de datos de video.

[0169] En algunos ejemplos, la finalización anticipada del proceso de búsqueda cuando la relación entre el costo mínimo global a través de la iteración de búsqueda actual y el costo mínimo a través de la iteración de búsqueda anterior está por debajo de un umbral de relación puede reducir la complejidad computacional sin aumentar materialmente la distorsión.

[0170] En algunos ejemplos, el proceso de búsqueda iterativa puede terminar anticipadamente en función de una relación entre el costo mínimo de la iteración de búsqueda anterior y la diferencia entre el costo mínimo de la iteración de búsqueda anterior y el costo mínimo global de la iteración de búsqueda actual en relación con un umbral de relación. En algunos ejemplos, un valor absoluto de la diferencia entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo global a través de la iteración de búsqueda actual puede ser utilizado en la ración. Por ejemplo, si la iteración de búsqueda actual es la 4ª iteración de búsqueda y la iteración de búsqueda anterior es la 3ª iteración de búsqueda, se determina una diferencia entre el costo mínimo a través de la 4ª iteración de búsqueda y el costo mínimo a través de la

3ª iteración de búsqueda, y se determina una relación entre la diferencia y el costo mínimo a través de la 3ª iteración de búsqueda. En algunos ejemplos, si la relación está por debajo de un umbral de relación, se termina el proceso de búsqueda iterativa. En algunos ejemplos, el umbral de relación puede ser fijo o estar en función de un tamaño del bloque de datos de video.

[0171] En algunos ejemplos, si el proceso de búsqueda iterativa no se termina anticipadamente y se produce otra iteración de búsqueda, el costo mínimo a través de la iteración de búsqueda actual se convierte en el costo mínimo a través de la iteración de búsqueda anterior para la siguiente iteración de búsqueda. Por ejemplo, si la iteración de búsqueda actual es la 4ª iteración de búsqueda y la iteración de búsqueda anterior es la 3ª iteración de búsqueda y el proceso de búsqueda iterativa no finaliza antes, se realiza una 5ª iteración de búsqueda. Para la 5ª iteración de búsqueda, el costo mínimo de la 4ª iteración de búsqueda se convierte en el costo mínimo de la iteración de búsqueda anterior y el costo mínimo global de la iteración de búsqueda actual se actualiza al costo mínimo de la 5ª iteración de búsqueda. En consecuencia, el costo mínimo global a través de la iteración de búsqueda actual y el costo mínimo a través de la iteración de búsqueda anterior pueden actualizarse para cada iteración de búsqueda.

[0172] En otro ejemplo, la etapa 3) sólo se aplica si el número de iteraciones anteriores supera un determinado umbral predefinido.

[0173] En otro ejemplo, el costo de la posición de búsqueda actual, denotado como `currCost`, puede ser considerado adicionalmente con `minCost` y `minCostPrev` para tomar una decisión sobre si terminar el proceso de búsqueda iterativa. Por ejemplo, se puede evaluar $\text{abs}(\text{currCost} - \text{minCostPrev})$, por ejemplo, comparando con umbrales, y si el costo actual `currCost` no es muy diferente de `minCostPrev`, se puede terminar la búsqueda. En otro ejemplo, se añade una condición adicional donde $\text{abs}(\text{minCostPrev} - \text{minCost})$ se compara con un umbral, y si es menor que el umbral y si $\text{abs}(\text{currCost} - \text{minCostPrev})$ es menor que otro umbral, entonces se termina el proceso de búsqueda iterativa.

[0174] Por ejemplo, si la iteración de búsqueda actual es la 4ª iteración de búsqueda y la iteración de búsqueda anterior es la 3ª iteración de búsqueda, el costo de la posición de búsqueda actual sería el costo de la propia 4ª iteración de búsqueda, mientras que un costo mínimo a través de una iteración de búsqueda anterior sería el costo mínimo a través de la 3ª iteración de búsqueda, y un costo mínimo global a través de una iteración de búsqueda actual sería el costo mínimo a través de la 4ª

iteración de búsqueda.

[0175] En algunos ejemplos, el codificador de video 200 y el decodificador de video 300 pueden estar configurados para comparar una diferencia entre un costo de la posición de búsqueda actual y el costo mínimo a través de la iteración de búsqueda anterior con un umbral de diferencia. En algunos ejemplos, el codificador de video 200 y el decodificador de video 300 pueden estar configurados para terminar el proceso de búsqueda iterativa anticipadamente si la diferencia entre un costo de la posición de búsqueda actual y el costo mínimo a través de la iteración de búsqueda anterior está por debajo de un umbral de diferencia.

[0176] En algunos ejemplos, el codificador de video 200 y el decodificador de video 300 pueden estar configurados para determinar una diferencia del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo global a través de la iteración de búsqueda actual y comparar la diferencia con un umbral de diferencia, como se ha comentado anteriormente. Además, el codificador de video 200 y el decodificador de video 300 pueden estar configurados para comparar una diferencia entre un costo de la posición de búsqueda actual y el costo mínimo a través de la iteración de búsqueda anterior con un umbral de diferencia. En algunos ejemplos, el codificador de video 200 y el decodificador de video 300 pueden estar configurados para terminar el proceso de búsqueda iterativa anticipadamente cuando ambas diferencias están por debajo de un umbral de diferencia respectivo. En algunos ejemplos, los umbrales de diferencia respectivos pueden ser iguales. En algunos ejemplos, los umbrales de diferencia respectivos pueden ser diferentes y/o independientes.

[0177] En algunos ejemplos, los umbrales de diferencia respectivos pueden ser fijo o en función de un tamaño del bloque de datos de video. En algunos ejemplos, los umbrales de diferencia respectivos pueden establecerse respectivamente igual a S veces el área del bloque de datos de video, donde S es un factor de escala que puede predefinirse o señalarse en el flujo de bits.

[0178] En algunos ejemplos, el codificador de video 200 y el decodificador de video 300 que terminan un proceso de búsqueda iterativa anticipadamente, de acuerdo con uno o más de los ejemplos analizados anteriormente, pueden reducir la complejidad computacional sin aumentar materialmente la distorsión.

[0179] LA Figura 9 es un diagrama de bloques que ilustra un codificador de video 200 de ejemplo que puede llevar a cabo las técnicas de esta divulgación. La Figura 9 se proporciona con fines explicativos y no se debe considerar limitativa de las técnicas de

acuerdo con lo que se ejemplifica y describe de manera más amplia en esta divulgación. Para fines de explicación, esta divulgación describe el codificador de video 200 de acuerdo con las técnicas de, VVC (ITU-T H.266, en desarrollo) y HEVC (ITU-T H.265). Sin embargo, las técnicas de esta divulgación pueden llevarse a cabo mediante dispositivos de codificación de video que están configurados para otros estándares de codificación de video y formatos de codificación de video, tales como AV1 y sucesores del formato de codificación de video AV1.

[0180] En el ejemplo de la Figura 9, el codificador de video 200 incluye la memoria de datos de video 230, la unidad de selección de modo 202, la unidad de generación residual 204, la unidad de procesamiento de transformada 206, la unidad de cuantificación 208, la unidad de cuantificación inversa 210, la unidad de procesamiento de transformada inversa 212, la unidad de reconstrucción 214, la unidad de filtro 216, el búfer de imagen decodificada (DPB) 218 y la unidad de codificación por entropía 220. Cualquiera o todos de la memoria de datos de video 230, la unidad de selección de modo 202, la unidad de generación residual 204, la unidad de procesamiento de transformada 206, la unidad de cuantificación 208, la unidad de cuantificación inversa 210, la unidad de procesamiento de transformada inversa 212, la unidad de reconstrucción 214, la unidad de filtro 216, el DPB 218 y la unidad de codificación por entropía 220 pueden implementarse en uno o más procesadores o en circuitería de procesamiento. Por ejemplo, las unidades del codificador de video 200 pueden implementarse como uno o más circuitos o elementos lógicos como parte de una circuitería de *hardware*, o como parte de un procesador, ASIC, o FPGA. Además, el codificador de video 200 puede incluir procesadores o circuitería de procesamiento adicionales o alternativos para llevar a cabo estas y otras funciones.

[0181] La memoria de datos de video 230 puede almacenar los datos de video que van a codificarse por los componentes del codificador de video 200. El codificador de video 200 puede recibir los datos de video almacenados en la memoria de datos de video 230 desde, por ejemplo, la fuente de video 104 (Figura 1). El DPB 218 puede actuar como una memoria de imágenes de referencia que almacena datos de video de referencia para su uso en la predicción de datos de video posteriores por el codificador de video 200. La memoria de datos de video 230 y el DPB 218 se pueden formar mediante cualquiera de una variedad de dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluso DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de video 230 y el DPB 218 pueden ser proporcionados

por el mismo dispositivo de memoria o por dispositivos de memoria separados. En diversos ejemplos, la memoria de datos de video 230 puede estar en un chip con otros componentes del codificador de video 200, como se ilustra, o fuera del chip con respecto a esos componentes.

[0182] En esta divulgación, la referencia a la memoria de datos de video 230 no debe interpretarse como limitada a la memoria interna del codificador de video 200, a menos que se describa específicamente como tal, o la memoria externa al codificador de video 200, a menos que se describa específicamente como tal. Más bien, la referencia a la memoria de datos de video 230 debe entenderse como una memoria de referencia que almacena datos de video que el codificador de video 200 recibe para codificar (por ejemplo, datos de video para un bloque actual que se va a codificar). La memoria 106 de la Figura 1 también puede proporcionar almacenamiento temporal de salidas de las diversas unidades del codificador de video 200.

[0183] Las diversas unidades de la Figura 9 se ilustran para ayudar a comprender las operaciones llevadas a cabo por el codificador de video 200. Las unidades pueden implementarse como circuitos de función fija, circuitos programables o una combinación de estos. Los circuitos de función fija se refieren a circuitos que proporcionan una funcionalidad particular y están preestablecidos en las operaciones que se pueden llevar a cabo. Los circuitos programables se refieren a los circuitos que se pueden programar para llevar a cabo diversas tareas y proporcionan una funcionalidad flexible en las operaciones que se pueden llevar a cabo. Por ejemplo, los circuitos programables pueden ejecutar *software* o *firmware* que hacen que los circuitos programables funcionen de la manera definida por las instrucciones del *software* o *firmware*. Los circuitos de función fija pueden ejecutar instrucciones de *software* (por ejemplo, para recibir parámetros o emitir parámetros), pero los tipos de operaciones que llevan a cabo los circuitos de función fija son generalmente inmutables. En algunos ejemplos, una o más de las unidades pueden ser bloques de circuitos distintos (de función fija o programables) y, en algunos ejemplos, una o más de las unidades pueden ser circuitos integrados.

[0184] El codificador de video 200 puede incluir unidades lógicas aritméticas (ALU), unidades de función elemental (EFU), circuitos digitales, circuitos analógicos y/o núcleos programables, formados a partir de circuitos programables. En los ejemplos donde las operaciones del codificador de video 200 se llevan a cabo mediante el uso de *software* ejecutado por los circuitos programables, la memoria 106 (Figura 1) puede almacenar instrucciones (por ejemplo, el código objeto) del *software* que el codificador

de video 200 recibe y ejecuta, u otra memoria dentro del codificador de video 200 (no mostrado) puede almacenar tales instrucciones.

[0185] La memoria de datos de video 230 está configurada para almacenar datos de video recibidos. El codificador de video 200 puede recuperar una imagen de los datos de video de la memoria de datos de video 230 y proporcionar los datos de video a la unidad de generación residual 204 y a la unidad de selección de modo 202. Los datos de video en la memoria de datos de video 230 pueden ser datos de video sin procesar que se van a codificar.

[0186] La unidad de selección de modo 202 incluye una unidad de estimación de movimiento 222, una unidad de compensación de movimiento 224 y una unidad de intrapredicción 226. La unidad de selección de modo 202 puede incluir unidades funcionales adicionales para llevar a cabo la predicción de video de acuerdo con otros modos de predicción. Como ejemplos, la unidad de selección de modo 202 puede incluir una unidad de paleta, una unidad de copia intrabloque (que puede ser parte de la unidad de estimación de movimiento 222 y/o la unidad de compensación de movimiento 224), una unidad afín, una unidad de modelo lineal (LM) o similar.

[0187] La unidad de selección de modo 202 generalmente coordina múltiples pasadas de codificación para probar combinaciones de parámetros de codificación y valores de distorsión de velocidad resultantes para tales combinaciones. Los parámetros de codificación pueden incluir la partición de CTU en CU, modos de predicción para las CU, tipos de transformación para datos residuales de las CU, parámetros de cuantificación para datos residuales de las CU, etc. La unidad de selección de modo 202 puede finalmente seleccionar la combinación de parámetros de codificación que tienen valores de distorsión de velocidad que son mejores que las otras combinaciones probadas.

[0188] El codificador de video 200 puede dividir una imagen recuperada de la memoria de datos de video 230 en una serie de CTU y encapsular una o más CTU dentro de una porción. La unidad de selección de modo 202 puede dividir una CTU de la imagen de acuerdo con una estructura de árbol, tal como la estructura de MTT, la estructura de QTBT, la estructura de superbloque o la estructura de árbol cuaternario descrita anteriormente. Como se describió anteriormente, el codificador de video 200 puede formar una o más CU a partir de la partición de una CTU de acuerdo con la estructura de árbol. Tal CU también puede denominarse generalmente "bloque de video" o "bloque".

[0189] En general, la unidad de selección de modo 202 también controla los

componentes de esta (por ejemplo, la unidad de estimación de movimiento 222, la unidad de compensación de movimiento 224 y la unidad de intrapredicción 226) para generar un bloque de predicción para un bloque actual (por ejemplo, una CU actual, o en HEVC, la porción superpuesta de una PU y una TU). Para la interpredicción de un bloque actual, la unidad de estimación de movimiento 222 puede llevar a cabo una búsqueda de movimiento para identificar uno o más bloques de referencia que coincidan estrechamente en una o más imágenes de referencia (por ejemplo, una o más imágenes codificadas previamente almacenadas en el DPB 218). En particular, la unidad de estimación de movimiento 222 puede calcular un valor representativo de qué tan similar es un bloque de referencia potencial al bloque actual, por ejemplo, de acuerdo con la suma de diferencia absoluta (SAD), la suma de diferencias cuadradas (SSD), la media de diferencia absoluta (MAD), la media de diferencias al cuadrado (MSD) o similar. Generalmente, la unidad de estimación de movimiento 222 puede llevar a cabo estos cálculos mediante el uso de las diferencias muestra por muestra entre el bloque actual y el bloque de referencia que se está considerando. La unidad de estimación de movimiento 222 puede identificar un bloque de referencia que tiene un valor más bajo resultante de estos cálculos, lo que indica un bloque de referencia que se asemeja más al bloque actual.

[0190] La unidad de estimación de movimiento 222 puede formar uno o más vectores de movimiento (MV) que definen las posiciones de los bloques de referencia en las imágenes de referencia con respecto a la posición del bloque actual en una imagen actual. La unidad de estimación de movimiento 222 puede formar un vector de movimiento mediante el uso de cualquiera de las técnicas para determinar un vector de movimiento, como se describió anteriormente. Por ejemplo, la unidad de estimación de movimiento 222 puede estar configurada para determinar un vector de movimiento para un bloque de datos de video mediante un proceso de derivación de vector de movimiento del lado del decodificador que incluye un proceso de búsqueda iterativa. El proceso de búsqueda iterativa incluye la determinación de un costo mínimo a través de una iteración de búsqueda anterior; la actualización de un costo mínimo global a través de una iteración de búsqueda actual; y la finalización anticipada del proceso de búsqueda iterativa en función de una comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo global a través de la iteración de búsqueda actual. La unidad de estimación de movimiento 222 puede estar configurada además para codificar el bloque de datos de video mediante el uso del vector de movimiento.

[0191] La unidad de estimación de movimiento 222 puede entonces proporcionar los vectores de movimiento a la unidad de compensación de movimiento 224. Por ejemplo, para la interpredicción unidireccional, la unidad de estimación de movimiento 222 puede proporcionar un único vector de movimiento, mientras que para la interpredicción bidireccional, la unidad de estimación de movimiento 222 puede proporcionar dos vectores de movimiento. La unidad de compensación de movimiento 224 puede generar entonces un bloque de predicción mediante los vectores de movimiento. Por ejemplo, la unidad de compensación de movimiento 224 puede recuperar datos del bloque de referencia mediante el vector de movimiento. Como otro ejemplo, si el vector de movimiento tiene una precisión de muestra fraccionada, la unidad de compensación de movimiento 224 puede interpolar valores para el bloque de predicción de acuerdo con uno o más filtros de interpolación. Además, para la interpredicción bidireccional, la unidad de compensación de movimiento 224 puede recuperar datos para dos bloques de referencia identificados por vectores de movimiento respectivos y combinar los datos recuperados, por ejemplo, mediante promedio muestra por muestra o promedio ponderado.

[0192] Al operar de acuerdo con el formato de codificación de video AV1, la unidad de estimación de movimiento 222 y la unidad de compensación de movimiento 224 pueden estar configuradas para codificar bloques de codificación de datos de video (por ejemplo, bloques de codificación de luminancia y crominancia) mediante el uso de compensación de movimiento traslacional, compensación de movimiento afín, compensación de movimiento por bloques superpuestos (OBMC), y/o interpredicción o intrapredicción compuesta.

[0193] Como otro ejemplo, para la intrapredicción o codificación de intrapredicción, la unidad de intrapredicción 226 puede generar el bloque de predicción a partir de muestras cercanas al bloque actual. Por ejemplo, para los modos direccionales, la unidad de intrapredicción 226 generalmente puede combinar matemáticamente valores de muestras cercanas y completar estos valores calculados en la dirección definida a través del bloque actual para producir el bloque de predicción. Como otro ejemplo, para el modo DC, la unidad de intrapredicción 226 puede calcular un promedio de las muestras cercanas al bloque actual y generar el bloque de predicción para incluir este promedio resultante para cada muestra del bloque de predicción.

[0194] Al operar de acuerdo con el formato de codificación de video AV1, se puede configurar la unidad de intrapredicción 226 para codificar bloques de codificación de datos de video (por ejemplo, bloques de codificación de luminancia y crominancia)

mediante el uso de intrapredicción direccional, intrapredicción no direccional, intrapredicción de filtro recursivo, predicción de crominancia a partir de luminancia (CFL), copia intrabloque (IBC) y/o modo de paleta de colores. La unidad de selección de modo 202 puede incluir unidades funcionales adicionales para llevar a cabo la predicción de video de acuerdo con otros modos de predicción.

[0195] La unidad de selección de modo 202 proporciona el bloque de predicción a la unidad de generación residual 204. La unidad de generación residual 204 recibe una versión en bruto, no codificada del bloque actual desde la memoria de datos de video 230 y el bloque de predicción desde la unidad de selección de modo 202. La unidad de generación residual 204 calcula las diferencias muestra por muestra entre el bloque actual y el bloque de predicción. Las diferencias de muestra por muestra resultantes definen un bloque residual para el bloque actual. En algunos ejemplos, la unidad de generación residual 204 también puede determinar diferencias entre valores de muestra en el bloque residual para generar un bloque residual mediante el uso de modulación por impulsos codificados diferencial residual (RDPCM). En algunos ejemplos, la unidad de generación residual 204 puede formarse mediante el uso de uno o más circuitos sustractores que llevan a cabo una resta binaria.

[0196] En los ejemplos donde la unidad de selección de modo 202 divide las CU en PU, cada PU puede estar asociada con una unidad de predicción de luminancia y las correspondientes unidades de predicción de crominancia. El codificador de video 200 y el decodificador de video 300 pueden admitir PU que tienen diversos tamaños. Como se indicó anteriormente, el tamaño de una CU se puede referir al tamaño del bloque de codificación de luminancia de la CU y el tamaño de una PU se puede referir al tamaño de una unidad de predicción de luminancia de la PU. Al suponer que el tamaño de una CU particular es $2N \times 2N$, el codificador de video 200 puede admitir tamaños de PU de $2N \times 2N$ o $N \times N$ para la intrapredicción y tamaños simétricos de PU de $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$ o similar para la interpredicción. El codificador de video 200 y el decodificador de video 300 también pueden admitir partición asimétrica para los tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$ para la interpredicción.

[0197] En los ejemplos en donde la unidad de selección de modo 202 no divide adicionalmente una CU en PU, cada CU puede estar asociada con un bloque de codificación de luminancia y los correspondientes bloques de codificación de crominancia. De acuerdo con lo anterior, el tamaño de una CU se puede referir al tamaño del bloque de codificación de luminancia de la CU. El codificador de video 200 y el decodificador de video 300 pueden admitir tamaños de CU de $2N \times 2N$, $2N \times N$ o

Nx2N.

[0198] Para otras técnicas de codificación de video, tal como una codificación de modo de copia intrabloque, una codificación de modo afín y una codificación de modo de modelo lineal (LM), como algunos ejemplos, la unidad de selección de modo 202, a través de las unidades respectivas asociadas con las técnicas de codificación, genera un bloque de predicción para el bloque actual que se está codificando. En algunos ejemplos, tal como la codificación del modo de paleta, es posible que la unidad de selección de modo 202 no genere un bloque de predicción y, en su lugar, genere elementos de sintaxis que indiquen la manera de reconstruir el bloque en función de una paleta seleccionada. En tales modos, la unidad de selección de modo 202 puede proporcionar estos elementos de sintaxis a la unidad de codificación por entropía 220 a codificar.

[0199] Como se describió anteriormente, la unidad de generación residual 204 recibe los datos de video para el bloque actual y el bloque de predicción correspondiente. La unidad de generación residual 204 luego genera un bloque residual para el bloque actual. Para generar el bloque residual, la unidad de generación residual 204 calcula las diferencias muestra por muestra entre el bloque de predicción y el bloque actual.

[0200] La unidad de procesamiento de transformación 206 aplica una o más transformaciones al bloque residual para generar un bloque de coeficientes de transformación (denominado en la presente "bloque de coeficientes de transformación"). La unidad de procesamiento de transformación 206 puede aplicar diversas transformaciones a un bloque residual para formar el bloque de coeficientes de transformación. Por ejemplo, la unidad de procesamiento de transformación 206 puede aplicar una transformación de coseno discreta (DCT), una transformación direccional, una transformación de Karhunen-Loeve (KLT) o una transformación conceptualmente similar a un bloque residual. En algunos ejemplos, la unidad de procesamiento de transformación 206 puede llevar a cabo múltiples transformaciones en un bloque residual, por ejemplo, una transformación primaria y una transformación secundaria, tal como una transformación rotacional. En algunos ejemplos, la unidad de procesamiento de transformación 206 no aplica transformaciones a un bloque residual.

[0201] Al operar según AV1, la unidad de procesamiento de transformación 206 aplica una o más transformaciones al bloque residual para generar un bloque de coeficientes de transformación (denominado en la presente "bloque de coeficientes de transformación"). La unidad de procesamiento de transformación 206 puede aplicar diversas transformaciones a un bloque residual para formar el bloque de coeficientes

de transformación. Por ejemplo, la unidad de procesamiento de transformaciones 206 puede aplicar una combinación de transformaciones horizontales/verticales que puede incluir una transformada discreta del coseno (DCT), una transformada discreta asimétrica del seno (ADST), una ADST invertida (por ejemplo, una ADST en orden inverso) y una transformada de identidad (IDTX). Cuando se utiliza una transformación de identidad, se saltea la transformación en una de las direcciones verticales u horizontales. En algunos ejemplos, se puede saltar el procesamiento de transformación.

[0202] La unidad de cuantificación 208 puede cuantificar los coeficientes de transformación en un bloque de coeficientes de transformación, para producir un bloque de coeficientes de transformación cuantificado. La unidad de cuantificación 208 puede cuantificar los coeficientes de transformación de un bloque de coeficientes de transformación de acuerdo con un valor de parámetro de cuantificación (QP) asociado con el bloque actual. El codificador de video 200 (por ejemplo, mediante la unidad de selección de modo 202) puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes de transformación asociados con el bloque actual al ajustar el valor de QP asociado con la CU. La cuantificación puede introducir pérdida de información y, por lo tanto, los coeficientes de transformación cuantificados pueden tener una precisión menor que los coeficientes de transformación originales producidos por la unidad de procesamiento de transformación 206.

[0203] La unidad de cuantificación inversa 210 y la unidad de procesamiento de transformación inversa 212 pueden aplicar cuantificación inversa y transformaciones inversas a un bloque de coeficientes de transformación cuantificado, respectivamente, para reconstruir un bloque residual a partir del bloque de coeficientes de transformación. La unidad de reconstrucción 214 puede producir un bloque reconstruido correspondiente al bloque actual (aunque potencialmente con algún grado de distorsión) en función del bloque residual reconstruido y un bloque de predicción generado por la unidad de selección de modo 202. Por ejemplo, la unidad de reconstrucción 214 puede agregar muestras del bloque residual reconstruido a las muestras correspondientes del bloque de predicción generado por la unidad de selección de modo 202 para producir el bloque reconstruido.

[0204] La unidad de filtro 216 puede llevar a cabo una o más operaciones de filtro en bloques reconstruidos. Por ejemplo, la unidad de filtro 216 puede llevar a cabo operaciones de desbloqueo para reducir los artefactos de bloqueo a lo largo de los bordes de las CU. Las operaciones de la unidad de filtro 216 pueden omitirse, en

algunos ejemplos.

[0205] Cuando se opera según AV1, la unidad de filtro 216 puede llevar a cabo una o más operaciones de filtro en bloques reconstruidos. Por ejemplo, la unidad de filtro 216 puede llevar a cabo operaciones de desbloqueo para reducir los artefactos de bloqueo a lo largo de los bordes de las CU. En otros ejemplos, la unidad de filtro 216 puede aplicar un filtro de mejora direccional restringido (CDEF), que puede aplicarse después del desbloqueo, y puede incluir la aplicación de filtros direccionales no separables, no lineales y de paso bajo en función de direcciones de borde estimadas. La unidad de filtrado 216 puede incluir además un filtro de restauración en bucle, lo cual se aplica luego de CDEF y puede incluir filtro Wiener normalizado simétrico y separable o un filtro dual auto guiado.

[0206] El codificador de video 200 almacena bloques reconstruidos en DPB 218. Por ejemplo, en los ejemplos en los que no se llevan a cabo las operaciones de la unidad de filtro 216, la unidad de reconstrucción 214 puede almacenar bloques reconstruidos en el DPB 218. En los ejemplos en los que se llevan a cabo las operaciones de la unidad de filtro 216, la unidad de filtro 216 puede almacenar los bloques reconstruidos filtrados en el DPB 218. La unidad de estimación de movimiento 222 y la unidad de compensación de movimiento 224 pueden recuperar una imagen de referencia del DPB 218, formada a partir de los bloques reconstruidos (y potencialmente filtrados), para interpredicir bloques de imágenes codificadas posteriormente. Además, la unidad de intrapredicción 226 puede usar bloques reconstruidos en el DPB 218 de una imagen actual para intrapredicir otros bloques en la imagen actual.

[0207] En general, la unidad de codificación por entropía 220 puede codificar por entropía elementos de sintaxis recibidos de otros componentes funcionales del codificador de video 200. Por ejemplo, la unidad de codificación por entropía 220 puede codificar por entropía bloques de coeficientes de transformación cuantificados a partir de la unidad de cuantificación 208. Como otro ejemplo, la unidad de codificación por entropía 220 puede codificar por entropía elementos de sintaxis de predicción (por ejemplo, información de movimiento para la interpredicción o información de intramodo para la intrapredicción) a partir de la unidad de selección de modo 202. La unidad de codificación por entropía 220 puede llevar a cabo una o más operaciones de codificación por entropía en los elementos de sintaxis, que son otro ejemplo de datos de video, para generar datos codificados por entropía. Por ejemplo, la unidad de codificación por entropía 220 puede llevar a cabo una operación de codificación adaptativa según el contexto de longitud variable (CAVLC), una operación de CABAC,

una operación de codificación de longitud de variable a variable (V2V), una operación de codificación aritmética binaria adaptativa según el contexto basado en sintaxis (SBAC), una operación de codificación por Entropía de partición de intervalo de probabilidad (PIPE), una operación de exponencial de codificación Golomb, u otro tipo de operación de codificación por entropía sobre los datos. En algunos ejemplos, la unidad de codificación por entropía 220 puede funcionar en modo de derivación donde los elementos de sintaxis no están codificados por entropía.

[0208] El codificador de video 200 puede generar un flujo de bits que incluye los elementos de sintaxis codificados por entropía necesarios para reconstruir bloques de un segmento o imagen. En particular, la unidad de codificación por entropía 220 puede generar el flujo de bits.

[0209] De acuerdo con AV1, la unidad de codificación de entropía 220 puede configurarse como un codificador aritmético multisímbolo adaptable símbolo a símbolo. Un elemento de sintaxis en AV1 incluye un alfabeto de elementos N y un contexto (por ejemplo, modelo de probabilidad) incluye un conjunto de probabilidades N. La unidad de codificación de entropía 220 puede almacenar las probabilidades como funciones de distribución acumulativa (CDF) de n bits (por ejemplo, 15 bits). La unidad de codificación de entropía 22 puede realizar un escalado recursivo, con un factor de actualización en función del tamaño del alfabeto, para actualizar los contextos.

[0210] Las operaciones que se describen anteriormente se describen con respecto a un bloque. Tal descripción debe entenderse como operaciones para un bloque de codificación de luminancia y/o bloques de codificación de crominancia. Como se describió anteriormente, en algunos ejemplos, el bloque de codificación de luminancia y los bloques de codificación de crominancia son componentes de luminancia y crominancia de una CU. En algunos ejemplos, el bloque de codificación de luminancia y los bloques de codificación de crominancia son componentes de luminancia y crominancia de una PU.

[0211] En algunos ejemplos, las operaciones llevadas a cabo con respecto a un bloque de codificación de luminancia no necesitan repetirse para los bloques de codificación de crominancia. Como ejemplo, las operaciones para identificar un vector de movimiento (MV) y una imagen de referencia para un bloque de codificación de luminancia no necesitan repetirse para identificar un MV y una imagen de referencia para los bloques de crominancia. Más bien, el MV para el bloque de codificación de luminancia se puede escalar para determinar el MV para los bloques de crominancia, y la imagen de referencia puede ser la misma. Como otro ejemplo, el proceso de

intrapredicción puede ser el mismo para el bloque de codificación de luminancia y los bloques de codificación de crominancia.

[0212] El codificador de video 200 representa un ejemplo de un dispositivo configurado para codificar datos de video que incluye una memoria configurada para almacenar datos de video, y una o más unidades de procesamiento implementadas en circuitería y configuradas para determinar un vector de movimiento para un bloque de datos de video mediante un proceso de derivación del vector de movimiento del lado del decodificador que incluye un proceso de búsqueda iterativa, en donde el proceso de búsqueda iterativa incluye: calcular un costo mínimo de una iteración de búsqueda anterior; y terminar anticipadamente el proceso de búsqueda iterativa en función del costo mínimo de la iteración de búsqueda anterior; y codificar el bloque de datos de video mediante el uso del vector de movimiento.

[0213] La Figura 10 es un diagrama de bloques que ilustra un decodificador de video 300 de ejemplo que puede realizar las técnicas de esta divulgación. La Figura 10 se proporciona con fines explicativos y no se limita a las técnicas que se ejemplifican y describen ampliamente en esta divulgación. Para fines de explicación, esta divulgación describe el decodificador de video 300 de acuerdo con las técnicas de VVC (ITU-T H.266, en desarrollo) y HEVC (ITU-T H.265). Sin embargo, las técnicas de esta divulgación pueden llevarse a cabo mediante dispositivos de codificación de video que están configurados para otros estándares de codificación de video.

[0214] En el ejemplo de la Figura 10, el decodificador de video 300 incluye la memoria de búfer de imagen codificada (CPB) 320, la unidad de decodificación por entropía 302, la unidad de procesamiento de predicción 304, la unidad de cuantificación inversa 306, la unidad de procesamiento de transformación inversa 308, la unidad de reconstrucción 310, la unidad de filtro 312 y el búfer de imagen decodificada (DPB) 314. Cualquiera o todos de la memoria de CPB 320, la unidad de decodificación por entropía 302, la unidad de procesamiento de predicción 304, la unidad de cuantificación inversa 306, la unidad de procesamiento de transformación inversa 308, la unidad de reconstrucción 310, la unidad de filtro 312 y el DPB 314 pueden implementarse en uno o más procesadores o en circuitería de procesamiento. Por ejemplo, las unidades del decodificador de video 300 pueden implementarse como uno o más circuitos o elementos lógicos como parte de una circuitería de *hardware*, o como parte de un procesador, ASIC, o FPGA. Además, el decodificador de video 300 puede incluir procesadores o circuitería de procesamiento adicionales o alternativos para llevar a cabo estas y otras funciones.

[0215] La unidad de procesamiento de predicción 304 incluye la unidad de compensación de movimiento 316 y la unidad de intrapredicción 318. La unidad de procesamiento de predicción 304 puede incluir unidades adicionales para llevar a cabo la predicción de acuerdo con otros modos de predicción. Como ejemplos, la unidad de procesamiento de predicción 304 puede incluir una unidad de paleta, una unidad de copia intrabloque (que puede formar parte de la unidad de compensación de movimiento 316), una unidad afín, una unidad de modelo lineal (LM) o similar. En otros ejemplos, el decodificador de video 300 puede incluir más, menos o diferentes componentes funcionales.

[0216] Al operar de acuerdo con AV1, la unidad de compensación de movimiento 316 puede estar configurada para codificar bloques de codificación de datos de video (por ejemplo, bloques de codificación de luminancia y crominancia) mediante el uso de compensación de movimiento traslacional, compensación de movimiento afín, OBMC, y/o interpredicción o intrapredicción compuesta, como se describió anteriormente. La unidad de intrapredicción 318 puede estar configurada para codificar los bloques de codificación de datos de video (por ejemplo, bloques de codificación de luminancia y crominancia) mediante el uso de intrapredicción direccional, intrapredicción no direccional, intrapredicción de filtro recursivo, CFL, copia intrabloque (IBC) y/o modo de paleta de colores, como se describió anteriormente.

[0217] La memoria de CPB 320 puede almacenar datos de video, tales como un flujo de bits de video codificado, para que los decodifiquen los componentes del decodificador de video 300. Los datos de video almacenados en la memoria de CPB 320 pueden obtenerse, por ejemplo, del medio legible por computadora 110 (Figura 1). La memoria de CPB 320 puede incluir un CPB que almacena datos de video codificados (por ejemplo, elementos de sintaxis) de un flujo de bits de video codificado. Además, la memoria de CPB 320 puede almacenar datos de video distintos de los elementos de sintaxis de una imagen codificada, tales como datos temporales que representan salidas de las diversas unidades del decodificador de video 300. La DPB 314 generalmente almacena imágenes decodificadas, cuyo decodificador de video 300 puede emitir y/o usar como datos de video de referencia cuando se decodifican datos o imágenes subsiguientes del flujo de bits de video codificado. La memoria de CPB 320 y el DPB 314 se pueden formar mediante cualquiera de una variedad de dispositivos de memoria, tal como DRAM, que incluye SDRAM, MRAM, RRAM u otros tipos de dispositivos de memoria. La memoria de CPB 320 y el DPB 314 pueden ser proporcionados por el mismo dispositivo de memoria o dispositivos de memoria

separados. En diversos ejemplos, la memoria de CPB 320 puede estar en un chip con otros componentes del decodificador de video 300, o puede estar fuera del chip con respecto a esos componentes.

[0218] Adicional o alternativamente, en algunos ejemplos, el decodificador de video 300 puede recuperar datos de video codificados de la memoria 120 (Figura 1). Es decir, la memoria 120 puede almacenar datos como se analizó anteriormente con respecto a la memoria CPB 320. Del mismo modo, la memoria 120 puede almacenar instrucciones que el decodificador de video 300 ejecutará, cuando parte o la totalidad de la funcionalidad del decodificador de video 300 se implementa en *software* para que la circuitería de procesamiento del decodificador de video 300 las ejecute.

[0219] Las diversas unidades que se muestran en la Figura 10 se ilustran para ayudar a comprender las operaciones llevadas a cabo por el decodificador de video 300. Las unidades pueden implementarse como circuitos de función fija, circuitos programables o una combinación de estos. De manera similar a la Figura 9, los circuitos de función fija se refieren a circuitos que proporcionan una funcionalidad particular y están preestablecidos en las operaciones que se pueden llevar a cabo. Los circuitos programables se refieren a los circuitos que se pueden programar para llevar a cabo diversas tareas y proporcionan una funcionalidad flexible en las operaciones que se pueden llevar a cabo. Por ejemplo, los circuitos programables pueden ejecutar *software* o *firmware* que hacen que los circuitos programables funcionen de la manera definida por las instrucciones del *software* o *firmware*. Los circuitos de función fija pueden ejecutar instrucciones de *software* (por ejemplo, para recibir parámetros o emitir parámetros), pero los tipos de operaciones que llevan a cabo los circuitos de función fija son generalmente inmutables. En algunos ejemplos, una o más de las unidades pueden ser bloques de circuitos distintos (de función fija o programables) y, en algunos ejemplos, una o más de las unidades pueden ser circuitos integrados.

[0220] El decodificador de video 300 puede incluir ALU, EFU, circuitos digitales, circuitos analógicos y/o núcleos programables formados a partir de circuitos programables. En los ejemplos donde las operaciones del decodificador de video 300 se llevan a cabo mediante *software* que se ejecuta en los circuitos programables, la memoria en el chip o fuera del chip puede almacenar instrucciones (por ejemplo, código objeto) del *software* que el decodificador de video 300 recibe y ejecuta.

[0221] La unidad de decodificación por entropía 302 puede recibir datos de video codificados desde el CPB y decodificar por entropía los datos de video para reproducir elementos de sintaxis. La unidad de procesamiento de predicción 304, la unidad de

cuantificación inversa 306, la unidad de procesamiento de transformación inversa 308, la unidad de reconstrucción 310, y la unidad de filtro 312 pueden generar datos de video decodificados en función de los elementos de sintaxis extraídos del flujo de bits.

[0222] En general, el decodificador de video 300 reconstruye una imagen bloque por bloque. El decodificador de video 300 puede llevar a cabo una operación de reconstrucción en cada bloque individualmente (donde el bloque que se está reconstruyendo actualmente, por ejemplo, decodificando, puede denominarse "bloque actual").

[0223] La unidad de decodificación por entropía 302 puede decodificar por entropía elementos de sintaxis que definen coeficientes de transformación cuantificados de un bloque de coeficientes de transformación cuantificados, así como información de transformación, tal como un parámetro de cuantificación (QP) y/o indicaciones de modo de transformación. La unidad de cuantificación inversa 306 puede usar el QP asociado con el bloque de coeficientes de transformación cuantificado para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa para que se aplique la unidad de cuantificación inversa 306. La unidad de cuantificación inversa 306 puede llevar a cabo, por ejemplo, una operación de desplazamiento a la izquierda bit a bit para cuantificar de forma inversa los coeficientes de transformación cuantificados. La unidad de cuantificación inversa 306 puede formar de ese modo un bloque de coeficientes de transformación que incluye coeficientes de transformación.

[0224] Después de que la unidad de cuantificación inversa 306 forme el bloque de coeficiente de transformación, la unidad de procesamiento de transformación inversa 308 puede aplicar una o más transformaciones inversas al bloque de coeficiente de transformación para generar un bloque residual asociado con el bloque actual. Por ejemplo, la unidad de procesamiento de transformación inversa 308 puede aplicar una DCT inversa, una transformación inversa de números enteros, una transformación inversa de Karhunen-Loeve (KLT), una transformación inversa de rotación, una transformación direccional inversa u otra transformación inversa al bloque de coeficientes de transformación.

Además, la unidad de procesamiento de predicción 304 genera un bloque de predicción de acuerdo con los elementos de sintaxis de información de predicción que fueron decodificados por entropía por la unidad de decodificación por entropía 302. Por ejemplo, si los elementos de sintaxis de información de predicción indican que el bloque actual está interpredicho, la unidad de compensación de movimiento 316 puede generar el bloque de predicción. En este caso, los elementos de sintaxis de

información de predicción pueden indicar una imagen de referencia en el DPB 314 de la cual recuperar un bloque de referencia, así como un vector de movimiento que identifica una ubicación del bloque de referencia en la imagen de referencia en relación con la ubicación del bloque actual en la imagen actual. La unidad de compensación de movimiento 316 generalmente puede llevar a cabo el proceso de interpredicción de una manera que es sustancialmente similar a la descrita con respecto a la unidad de compensación de movimiento 224 (Figura 9).

[0225] La unidad de compensación de movimiento 316 puede formar un vector de movimiento mediante el uso de cualquiera de las técnicas para determinar un vector de movimiento, como se describió anteriormente. Por ejemplo, la unidad de compensación de movimiento 316 puede estar configurada para determinar un vector de movimiento para un bloque de datos de video mediante un proceso de derivación de vector de movimiento del lado del decodificador que incluye un proceso de búsqueda iterativa. El proceso de búsqueda iterativa incluye la determinación de un costo mínimo a través de una iteración de búsqueda anterior; la actualización de un costo mínimo global a través de una iteración de búsqueda actual; y la finalización anticipada del proceso de búsqueda iterativa en función de una comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo global a través de la iteración de búsqueda actual. La unidad de compensación de movimiento 316 puede estar configurada además para decodificar el bloque de datos de video mediante el uso del vector de movimiento.

[0226] Como otro ejemplo, si los elementos de sintaxis de información de predicción indican que el bloque actual está intrapredicho, la unidad de intrapredicción 318 puede generar el bloque de predicción de acuerdo con un modo de intrapredicción indicado por los elementos de sintaxis de información de predicción. De nuevo, la unidad de intrapredicción 318 generalmente puede llevar a cabo el proceso de intrapredicción de una manera que es sustancialmente similar a la descrita con respecto a la unidad de intrapredicción 226 (Figura 9). La unidad de intrapredicción 318 puede recuperar datos de muestras cercanas al bloque actual del DPB 314.

[0227] La unidad de reconstrucción 310 puede reconstruir el bloque actual mediante el uso del bloque de predicción y el bloque residual. Por ejemplo, la unidad de reconstrucción 310 puede agregar muestras del bloque residual a las muestras correspondientes del bloque de predicción para reconstruir el bloque actual.

[0228] La unidad de filtro 312 puede llevar a cabo una o más operaciones de filtro en bloques reconstruidos. Por ejemplo, la unidad de filtro 312 puede llevar a cabo

operaciones de desbloqueo para reducir los artefactos de bloqueo a lo largo de los bloques reconstruidos. Las operaciones de la unidad de filtro 312 no se llevan a cabo necesariamente en todos los ejemplos.

[0229] El decodificador de video 300 puede almacenar los bloques reconstruidos en el DPB 314. Por ejemplo, en los ejemplos donde no se llevan a cabo las operaciones de la unidad de filtro 312, la unidad de reconstrucción 310 puede almacenar bloques reconstruidos en el DPB 314. En los ejemplos donde se llevan a cabo las operaciones de la unidad de filtro 312, la unidad de filtro 312 puede almacenar los bloques reconstruidos filtrados en el DPB 314. Como se analiza anteriormente, el DPB 314 puede proporcionar información de referencia, tal como muestras de una imagen actual para intrapredicción e imágenes previamente decodificadas para una compensación de movimiento posterior, a la unidad de procesamiento de predicción 304. Además, el decodificador de video 300 puede emitir imágenes decodificadas (por ejemplo, video decodificado) desde el DPB 314 para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 118 de la Figura 1.

[0230] De esta manera, el decodificador de video 300 representa un ejemplo de un dispositivo de decodificación de video que incluye una memoria configurada para almacenar datos de video, y una o más unidades de procesamiento implementadas en circuitería y configuradas para determinar un vector de movimiento para un bloque de datos de video mediante un proceso de derivación del vector de movimiento del lado del decodificador que incluye un proceso de búsqueda iterativa, en donde el proceso de búsqueda iterativa incluye: calcular un costo mínimo de una iteración de búsqueda anterior; y terminar anticipadamente el proceso de búsqueda iterativa en función del costo mínimo de la iteración de búsqueda anterior; y codificar el bloque de datos de video mediante el uso del vector de movimiento.

[0231] La Figura 11 es un diagrama de flujo que ilustra un método de ejemplo para codificar un bloque actual de acuerdo con las técnicas de esta divulgación. El bloque actual puede comprender una CU actual. Aunque se describe con respecto al codificador de video 200 (Figuras 1 y 9), debe entenderse que se pueden configurar otros dispositivos para realizar un método similar al de la Figura 11.

[0232] En este ejemplo, el codificador de video 200 predice inicialmente el bloque actual (350). Por ejemplo, el codificador de video 200 puede formar un bloque de predicción para el bloque actual. El codificador de video 200 luego puede calcular un bloque residual para el bloque actual (352). Para calcular el bloque residual, el codificador de video 200 puede calcular una diferencia entre el bloque original no

codificado y el bloque de predicción para el bloque actual. El codificador de video 200 luego puede transformar el bloque residual y cuantificar los coeficientes de transformación del bloque residual (354). Luego, el codificador de video 200 puede explorar los coeficientes de transformación cuantificados del bloque residual (356). Durante la exploración, o después de la exploración, el codificador de video 200 puede codificar por entropía los coeficientes de transformación (358). Por ejemplo, el codificador de video 200 puede codificar los coeficientes de transformación mediante el uso de CAVLC o CABAC. El codificador de video 200 puede generar entonces los datos codificados por entropía del bloque (360).

[0233] La Figura 12 es un diagrama de flujo que ilustra un método de ejemplo para decodificar un bloque actual de datos de video de acuerdo con las técnicas de esta divulgación. El bloque actual puede comprender una CU actual. Aunque se describe con respecto al decodificador de video 300 (Figuras 1 y 10), debe entenderse que se pueden configurar otros dispositivos para realizar un método similar al de la Figura 12.

[0234] El decodificador de video 300 puede recibir datos codificados por entropía para el bloque actual, tales como información de predicción codificada por entropía y datos codificados por entropía para coeficientes de transformación de un bloque residual correspondiente al bloque actual (370). El decodificador de video 300 puede decodificar por entropía los datos codificados por entropía para determinar la información de predicción para el bloque actual y reproducir los coeficientes de transformación del bloque residual (372). El decodificador de video 300 puede predecir el bloque actual (374), por ejemplo, mediante el uso de un modo de intrapredicción o interpredicción, como lo indica la información de predicción para el bloque actual, para calcular un bloque de predicción para el bloque actual. El decodificador de video 300 luego puede explorar de forma inversa los coeficientes de transformación reproducidos (376), para crear un bloque de coeficientes de transformación cuantificados. El decodificador de video 300 puede entonces cuantificar inversamente los coeficientes de transformación y aplicar una transformación inversa a los coeficientes de transformación para producir un bloque residual (378). El decodificador de video 300 puede finalmente decodificar el bloque actual al combinar el bloque de predicción y el bloque residual (380).

[0235] La Figura 13 es un diagrama de flujo que ilustra un método de ejemplo para terminar anticipadamente un proceso de búsqueda iterativa de acuerdo con las técnicas de esta divulgación. El bloque actual puede comprender una CU actual. El proceso de búsqueda iterativa se analiza con respecto a su realización por el

codificador de video 200 y/o el decodificador de video 300.

[0236] El codificador de video 200/decodificador de video 300 puede determinar un vector de movimiento para un bloque de datos de video mediante un proceso de derivación de vector de movimiento del lado del decodificador que incluye un proceso de búsqueda iterativa. En un ejemplo, en el proceso de búsqueda iterativa, el codificador de video 200/decodificador de video 300 puede configurarse para realizar una primera iteración de búsqueda y establecer un costo mínimo a través de una iteración de búsqueda anterior en cero (1300). El codificador de video 200/decodificador de video 300 puede configurarse además para determinar un costo a través de la primera iteración de búsqueda y establecer un costo mínimo general a través de la iteración de búsqueda actual como el costo total a través de la primera iteración de búsqueda (1302). El codificador de video 200/decodificador de video 300 puede configurarse además para determinar si se debe terminar el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de una iteración de búsqueda anterior con el costo mínimo general a través de la iteración de búsqueda actual (1304). Si el codificador de video 200/decodificador de video 300 determina terminar el proceso de búsqueda iterativa anticipadamente, el codificador de video 200/decodificador de video 300 se configura además para codificar/decodificar respectivamente el bloque de datos de video mediante el uso del vector de movimiento que tiene el costo mínimo hasta ese punto en el proceso de búsqueda (1305).

[0237] Si el codificador de video 200/decodificador de video 300 determina no terminar el proceso de búsqueda iterativa anticipadamente, el codificador de video 200/decodificador de video 300 se configura además para realizar una segunda iteración de búsqueda y establecer un costo mínimo a través de una iteración de búsqueda anterior como el costo mínimo a través de la primera iteración de búsqueda (1306). El codificador de video 200/decodificador de video 300 puede configurarse además para actualizar un costo mínimo general a través de la iteración de búsqueda actual como el costo total a través de la segunda iteración de búsqueda (1308). El codificador de video 200/decodificador de video 300 puede configurarse además para determinar si se debe terminar el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de una iteración de búsqueda anterior con el costo mínimo general a través de la iteración de búsqueda actual (1310). Si el codificador de video 200/decodificador de video 300 determina terminar el proceso de búsqueda iterativa anticipadamente, el codificador de video 200/decodificador de video 300 se configura además para codificar/decodificar respectivamente el bloque de

datos de video mediante el uso del vector de movimiento que tiene el costo mínimo hasta ese punto en el proceso de búsqueda (1305).

[0238] Si el codificador de video 200/decodificador de video 300 determina no terminar el proceso de búsqueda iterativa anticipadamente, el codificador de video 200/decodificador de video 300 se configura además para realizar una tercera iteración de búsqueda y establecer un costo mínimo a través de una iteración de búsqueda anterior como el costo mínimo a través de la segunda iteración de búsqueda (1312). El codificador de video 200/decodificador de video 300 puede configurarse además para actualizar un costo mínimo general a través de la iteración de búsqueda actual como el costo total a través de la tercera iteración de búsqueda (1314). El codificador de video 200/decodificador de video 300 puede configurarse además para determinar si se debe terminar el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de una iteración de búsqueda anterior con el costo mínimo general a través de la iteración de búsqueda actual (1316). Si el codificador de video 200/decodificador de video 300 determina terminar el proceso de búsqueda iterativa anticipadamente, el codificador de video 200/decodificador de video 300 se configura además para codificar/decodificar respectivamente el bloque de datos de video mediante el uso del vector de movimiento que tiene el costo mínimo hasta ese punto en el proceso de búsqueda (1305).

[0239] Si el codificador de video 200/decodificador de video 300 determina no terminar el proceso de búsqueda iterativa anticipadamente, el codificador de video 200/decodificador de video 300 se configura además para realizar una siguiente iteración de búsqueda de acuerdo con la iteración de búsqueda anterior hasta que el codificador de video 200/decodificador de video 300 termine el proceso de búsqueda iterativa (1318).

[0240] En algunos ejemplos, terminar el proceso de búsqueda iterativa anticipadamente en función de una comparación del costo mínimo a través de la iteración de búsqueda previa y el costo mínimo general a través de la iteración de búsqueda actual puede realizarse en función de múltiples comparaciones.

[0241] La Figura 14 es un diagrama de flujo que ilustra otro ejemplo de método para la decodificación de un bloque de datos de video de acuerdo con las técnicas de esta divulgación. Las técnicas de la Figura 14 puede realizarse mediante una o más componentes estructurales del decodificador de video 300, que incluye la unidad de compensación de movimiento 316.

[0242] En un ejemplo de la divulgación, el decodificador de video 300 puede estar

configurada para determinar un vector de movimiento para un bloque de datos de video mediante un proceso de derivación de vector de movimiento del lado del decodificador que incluye un proceso de búsqueda iterativa (1400). En el proceso de búsqueda iterativa, el decodificador de video 300 puede configurarse además para determinar un costo mínimo a través de una iteración de búsqueda anterior (1402), actualizar de un costo mínimo global a través de una iteración de búsqueda actual (1404), finalizar anticipadamente el proceso de búsqueda iterativa en función de una comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo global a través de la iteración de búsqueda actual (1406). El decodificador de video 300 puede entonces decodificar un bloque de datos de video mediante el uso del vector de movimiento (1408).

[0243] En un ejemplo, el decodificador de video 300 puede configurarse para calcular el costo mínimo general a través de la iteración de búsqueda actual en función de al menos un subconjunto actual de posiciones candidatas.

[0244] En otro ejemplo, para terminar anticipadamente el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, el decodificador de video 300 puede configurarse para calcular una diferencia entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, y finalizar el proceso de búsqueda iterativa anticipadamente en función de la diferencia relativa a un umbral de diferencia.

[0245] En otro ejemplo, para terminar anticipadamente el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, el decodificador de video 300 puede configurarse para calcular una relación entre costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, y finalizar el proceso de búsqueda iterativa anticipadamente en función de la relación relativa a un umbral de relación.

[0246] En otro ejemplo, para terminar anticipadamente el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, el decodificador de video 300 puede configurarse para calcular una diferencia entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, calcular una relación entre costo mínimo a través de la iteración de búsqueda anterior y la diferencia entre el costo

mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, y finalizar el proceso de búsqueda iterativa anticipadamente en función de la relación relativa a un umbral de relación.

[0247] En otro ejemplo, el decodificador de video 300 puede configurarse para terminar el proceso de búsqueda iterativa anticipadamente en función del costo mínimo a través de la iteración de búsqueda previa, el costo mínimo general a través de la iteración de búsqueda actual, y un costo mínimo de la iteración de búsqueda actual. Por ejemplo, el decodificador de video 300 puede configurarse para terminar el proceso de búsqueda iterativa anticipadamente en función de la comparación entre el costo mínimo a través de la iteración de búsqueda previa y el costo mínimo general a través de la iteración de búsqueda actual, y una comparación entre el costo mínimo a través de la iteración de búsqueda previa y el costo mínimo de la iteración de búsqueda actual.

[0248] La Figura 15 es un diagrama de flujo que ilustra otro ejemplo de método para la codificación de un bloque de datos de video de acuerdo con las técnicas de esta divulgación. Las técnicas de la Figura 15 pueden realizarse mediante una o más componentes estructurales del codificador de video 200, que incluye la unidad de estimación de movimiento 222 y/o la unidad de compensación de movimiento 224.

[0249] En un ejemplo de la divulgación, el codificador de video 200 puede estar configurada para determinar un vector de movimiento para un bloque de datos de video mediante un proceso de derivación de vector de movimiento del lado del decodificador que incluye un proceso de búsqueda iterativa (1500). En el proceso de búsqueda iterativa, el decodificador de video 300 puede configurarse además para determinar un costo mínimo a través de una iteración de búsqueda anterior (1502), actualizar de un costo mínimo global a través de una iteración de búsqueda actual (1504), finalizar anticipadamente el proceso de búsqueda iterativa en función de una comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo global a través de la iteración de búsqueda actual (1506). El codificador de video 200 puede entonces codificar un bloque de datos de video mediante el uso del vector de movimiento (1508).

[0250] En un ejemplo, el codificador de video 200 puede configurarse para calcular el costo mínimo general a través de la iteración de búsqueda actual en función de al menos un subconjunto actual de posiciones candidatas.

[0251] En otro ejemplo, para terminar anticipadamente el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de la iteración de

búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, el codificador de video 200 puede configurarse para calcular una diferencia entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, y finalizar el proceso de búsqueda iterativa anticipadamente en función de la diferencia relativa a un umbral de diferencia.

[0252] En otro ejemplo, para terminar anticipadamente el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, el codificador de video 200 puede configurarse para calcular una relación entre costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, y finalizar el proceso de búsqueda iterativa anticipadamente en función de la relación relativa a un umbral de relación.

[0253] En otro ejemplo, para terminar anticipadamente el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, el codificador de video 200 puede configurarse para calcular una diferencia entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, calcular una relación entre costo mínimo a través de la iteración de búsqueda anterior y la diferencia entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, y finalizar el proceso de búsqueda iterativa anticipadamente en función de la relación relativa a un umbral de relación.

[0254] En otro ejemplo, el codificador de video 200 puede configurarse para terminar el proceso de búsqueda iterativa anticipadamente en función del costo mínimo a través de la iteración de búsqueda previa, el costo mínimo general a través de la iteración de búsqueda actual, y un costo mínimo de la iteración de búsqueda actual. Por ejemplo, el decodificador de video 200 puede configurarse para terminar el proceso de búsqueda iterativa anticipadamente en función de la comparación entre el costo mínimo a través de la iteración de búsqueda previa y el costo mínimo general a través de la iteración de búsqueda actual, y una comparación entre el costo mínimo a través de la iteración de búsqueda previa y el costo mínimo de la iteración de búsqueda actual.

[0255] El método y los dispositivos adicionales de la divulgación se describen en los siguientes aspectos.

[0256] Aspecto 1A - Un método de codificación de datos de video que comprende:

determinar un vector de movimiento para un bloque de datos de video mediante un proceso de derivación del vector de movimiento del lado del decodificador que incluye un proceso de búsqueda iterativa, en donde el proceso de búsqueda iterativa incluye: calcular un costo mínimo de una iteración de búsqueda anterior; y terminar anticipadamente el proceso de búsqueda iterativa en función del costo mínimo de la iteración de búsqueda anterior; y codificar el bloque de datos de video mediante el uso del vector de movimiento.

[0257] Aspecto 2A - El método de acuerdo con el aspecto 1A, que comprende además: actualizar un costo mínimo determinado para el proceso de búsqueda iterativa.

[0258] Aspecto 3A - El método de acuerdo con el aspecto 2A, en donde terminar anticipadamente el proceso de búsqueda iterativa en función del costo mínimo de la iteración de búsqueda anterior comprende: calcular una diferencia o una proporción del costo mínimo de la iteración de búsqueda anterior y el costo mínimo determinado para el proceso de búsqueda iterativa; y terminar el proceso de búsqueda iterativa anticipadamente en función de la diferencia o la proporción relativa a un umbral.

[0259] Aspecto 4A - El método de acuerdo con el aspecto 2A, en donde terminar anticipadamente el proceso de búsqueda iterativa en función del costo mínimo de la iteración de búsqueda anterior comprende: terminar el proceso de búsqueda iterativa anticipadamente en función del costo mínimo de la iteración de búsqueda anterior, el costo mínimo determinado para el proceso de búsqueda iterativa y un costo mínimo de una iteración actual.

[0260] Aspecto 5A - El método de acuerdo con el aspecto 1A, en donde el proceso de búsqueda iterativa incluye más de dos iteraciones.

[0261] Aspecto 6A - El método de acuerdo con el aspecto 1A, en donde el proceso de búsqueda iterativa incluye coincidencia bilateral o comparación de plantillas.

[0262] Aspecto 7A - El método de acuerdo con cualquiera de los Aspectos 1A a 6A, en donde la codificación comprende la decodificación.

[0263] Aspecto 8A - El método de acuerdo con cualquiera de los Aspectos 1A a 6A, en donde la codificación comprende la codificación.

[0264] Aspecto 9A - Un dispositivo para codificar datos de video, en donde el dispositivo comprende uno o más medios para llevar a cabo el método de acuerdo con cualquiera de los Aspectos 1A–8A.

[0265] Aspecto 10A - El dispositivo de acuerdo con el aspecto 9A, en donde el uno o más medios comprenden uno o más procesadores implementados en circuitería.

[0266] Aspecto 11A - El dispositivo de acuerdo con cualquiera de los aspectos 9A y 10A, que comprende además una memoria para almacenar los datos de video.

[0267] Aspecto 12A - El dispositivo de acuerdo con cualquiera de los Aspectos 9A-11A, que comprende además una pantalla configurada para mostrar datos de video decodificados.

[0268] Aspecto 13A - El dispositivo de acuerdo con cualquiera de los Aspectos 9A-12A, en donde el dispositivo comprende uno o más de una cámara, una computadora, un dispositivo móvil, un dispositivo receptor de transmisión o un decodificador.

[0269] Aspecto 14A - El dispositivo de acuerdo con cualquiera de los aspectos 9A-13A, en donde el dispositivo comprende un decodificador de video.

[0270] Aspecto 15A - El dispositivo de acuerdo con cualquiera de los aspectos 9A-14A, en donde el dispositivo comprende un codificador de video.

[0271] Aspecto 16A - Un medio de almacenamiento legible por computadora que tiene almacenadas instrucciones que, cuando se ejecutan, hacen que uno o más procesadores lleven a cabo el método de acuerdo con cualquiera de los aspectos 1A-8A.

[0272] Aspecto 17A - Un dispositivo para codificar datos de video, que comprende: medios para determinar un vector de movimiento para un bloque de datos de video mediante un proceso de derivación del vector de movimiento del lado del decodificador que incluye un proceso de búsqueda iterativa, en donde el proceso de búsqueda iterativa incluye: calcular un costo mínimo de una iteración de búsqueda anterior; y terminar anticipadamente el proceso de búsqueda iterativa en función del costo mínimo de la iteración de búsqueda anterior; y medios para codificar el bloque de datos de video mediante el uso del vector de movimiento.

[0273] Aspecto 18A - Un método de codificación de datos de video que comprende: determinar un vector de movimiento para un bloque de datos de video mediante un proceso de derivación del vector de movimiento del lado del decodificador que incluye un proceso de búsqueda iterativa, en donde el proceso de búsqueda iterativa incluye: calcular un costo mínimo de una iteración de búsqueda anterior; y terminar anticipadamente el proceso de búsqueda iterativa en función del costo mínimo de la iteración de búsqueda anterior; y codificar el bloque de datos de video mediante el uso del vector de movimiento.

[0274] Aspecto 19A - El método de acuerdo con el aspecto 18A, que comprende además: actualizar un costo mínimo determinado para el proceso de búsqueda iterativa.

[0275] Aspecto 20A - El método de acuerdo con el aspecto 19A, en donde terminar anticipadamente el proceso de búsqueda iterativa en función del costo mínimo de la iteración de búsqueda anterior comprende: calcular una diferencia o una proporción del costo mínimo de la iteración de búsqueda anterior y el costo mínimo determinado para el proceso de búsqueda iterativa; y terminar el proceso de búsqueda iterativa anticipadamente en función de la diferencia o la proporción relativa a un umbral.

[0276] Aspecto 21A - El método de acuerdo con el aspecto 19A, en donde terminar anticipadamente el proceso de búsqueda iterativa en función del costo mínimo de la iteración de búsqueda anterior comprende: terminar el proceso de búsqueda iterativa anticipadamente en función del costo mínimo de la iteración de búsqueda anterior, el costo mínimo determinado para el proceso de búsqueda iterativa y un costo mínimo de una iteración actual.

[0277] Aspecto 22A - El método de acuerdo con cualquiera de los aspectos 18A-21A, en donde el proceso de búsqueda iterativa incluye más de dos iteraciones.

[0278] Aspecto 23A - El método de acuerdo con cualquiera de los aspectos 18A-22A, en donde el proceso de búsqueda iterativa incluye coincidencia bilateral o comparación de plantillas.

[0279] Aspecto 24A - Un dispositivo para codificar datos de video, en donde el dispositivo comprende uno o más medios para llevar a cabo el método de acuerdo con cualquiera de los Aspectos 18A-23A.

[0280] Aspecto 1B - Un método para codificar o decodificar datos de video incluye determinar un vector de movimiento para un bloque de datos de video mediante el uso de un proceso de derivación del vector de movimiento del lado del decodificador que incluye llevar a cabo un proceso de búsqueda iterativa, en donde llevar a cabo el proceso de búsqueda iterativa incluye: determinar un costo mínimo a través de una iteración de búsqueda anterior; actualizar el costo mínimo general a través de una iteración de búsqueda actual; y terminar anticipadamente el proceso de búsqueda iterativa en función de una comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, y codificar o decodificar el bloque de datos de video mediante el uso del vector de movimiento.

[0281] Aspecto 2B - El método de acuerdo con el aspecto 1B, en donde la iteración de búsqueda anterior precede inmediatamente a la iteración de búsqueda actual en el proceso de búsqueda iterativa.

[0282] Aspecto 3B - El método de acuerdo con el aspecto 1B incluye además calcular

el costo mínimo general a través de la iteración de búsqueda actual en función de al menos un subconjunto actual de posiciones candidatas.

[0283] Aspecto 4B - El método de acuerdo con el aspecto 1B, en donde terminar anticipadamente el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual comprende calcular una diferencia entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual; y finalizar el proceso de búsqueda iterativa anticipadamente en función de la diferencia relativa a un umbral de diferencia.

[0284] Aspecto 5B - El método de acuerdo con el aspecto 4B, en donde el umbral de diferencia se basa en el tamaño del bloque.

[0285] Aspecto 6B - El método de acuerdo con el aspecto 1B, en donde terminar anticipadamente el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual comprende calcular una relación entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual; y finalizar el proceso de búsqueda iterativa anticipadamente en función de la relación relativa a un umbral de relación.

[0286] Aspecto 7B - El método de acuerdo con el aspecto 6B, en donde el umbral de relación es fijo o se basa en el tamaño del bloque.

[0287] Aspecto 8B - El método de acuerdo con el aspecto 1B, en donde terminar anticipadamente el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual comprende: calcular una diferencia entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual; calcular una relación entre costo mínimo a través de la iteración de búsqueda anterior y la diferencia entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual; y finalizar el proceso de búsqueda iterativa anticipadamente en función de la relación relativa a un umbral de relación.

[0288] Aspecto 9B - El método de acuerdo con el aspecto 8B, en donde el umbral de relación es fijo o se basa en el tamaño del bloque.

[0289] Aspecto 10B - El método de acuerdo con el aspecto 1B, en donde terminar anticipadamente el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a

través de la iteración de búsqueda actual comprende: terminar el proceso de búsqueda iterativa anticipadamente en función de la comparación entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual y una comparación entre el costo mínimo a través de la iteración de búsqueda anterior y un costo mínimo de la iteración de búsqueda actual.

[0290] Aspecto 11B - El método de acuerdo con el aspecto 1B, en donde el proceso de búsqueda iterativa incluye más de dos iteraciones.

[0291] Aspecto 12B - El método de acuerdo con el aspecto 1B, en donde realizar el proceso de búsqueda iterativa incluye realizar una coincidencia bilateral o una comparación de plantillas.

[0292] Aspecto 13B - Un dispositivo para procesar datos de video incluye una memoria configurada para almacenar los datos de video; y una circuitería de procesamiento configurada para: determinar un vector de movimiento para un bloque de los datos de video mediante un proceso de derivación del vector de movimiento del lado del decodificador que incluye un proceso de búsqueda iterativa, en donde, para realizar el proceso de búsqueda iterativa, el circuito de procesamiento está además configurado para: determinar un costo mínimo a través de una iteración de búsqueda anterior; actualizar el costo mínimo general a través de una iteración de búsqueda actual; y terminar anticipadamente el proceso de búsqueda iterativa en función de una comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, y codificar o decodificar el bloque de datos de video mediante el uso del vector de movimiento.

[0293] Aspecto 14B - El dispositivo de acuerdo con el aspecto 13B, en donde la iteración de búsqueda anterior precede inmediatamente a la iteración de búsqueda actual.

[0294] Aspecto 15B - El dispositivo del aspecto 13B, en donde, para realizar el proceso de búsqueda iterativa, el circuito de procesamiento está configurado además para calcular el costo mínimo general a través de la iteración de búsqueda actual en función de al menos un subconjunto actual de posiciones candidatas.

[0295] Aspecto 16B - El dispositivo de acuerdo con el aspecto 13B, en donde, para terminar anticipadamente el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, el circuito de procesamiento se configura, además, para: calcular una diferencia entre el costo

mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual; y finalizar el proceso de búsqueda iterativa anticipadamente en función de la diferencia relativa a un umbral de diferencia.

[0296] Aspecto 17B - El dispositivo de acuerdo con el aspecto 16B, en donde el umbral de diferencia se basa en el tamaño del bloque.

[0297] Aspecto 18B - El dispositivo de acuerdo con el aspecto 13B, en donde, para terminar anticipadamente el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, el circuito de procesamiento se configura, además, para: calcular una relación entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual; y finalizar el proceso de búsqueda iterativa anticipadamente en función de la relación relativa a un umbral de relación.

[0298] Aspecto 19B - El dispositivo de acuerdo con el aspecto 18B, en donde el umbral de relación es fijo o se basa en el tamaño del bloque.

[0299] Aspecto 20B - El dispositivo de acuerdo con el aspecto 13B, en donde para terminar anticipadamente el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, el circuito de procesamiento se configura además para: calcular una diferencia entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual; calcular una relación entre costo mínimo a través de la iteración de búsqueda anterior y la diferencia entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual; y finalizar el proceso de búsqueda iterativa anticipadamente en función de la relación relativa a un umbral de relación.

[0300] Aspecto 21B - El dispositivo de acuerdo con el aspecto 20B, en donde el umbral de relación es fijo o se basa en el tamaño del bloque.

[0301] Aspecto 22B - El dispositivo de acuerdo con el aspecto 13B, en donde para terminar anticipadamente el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, el circuito de procesamiento se configura además para: terminar el proceso de búsqueda iterativa anticipadamente en función de la comparación entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de

búsqueda actual y una comparación entre el costo mínimo a través de la iteración de búsqueda anterior y un costo mínimo de la iteración de búsqueda actual.

[0302] Aspecto 23B - El dispositivo de acuerdo con el aspecto 13B, en donde el proceso de búsqueda iterativa incluye más de dos iteraciones.

[0303] Aspecto 24B - El dispositivo de acuerdo con el aspecto 13B, en donde, para realizar el proceso de búsqueda iterativa, el circuito de procesamiento se configura para realizar una coincidencia bilateral o una comparación de plantillas.

[0304] Aspecto 25B - El dispositivo de acuerdo con el aspecto 13B, que comprende además una pantalla configurada para mostrar los datos de video.

[0305] Aspecto 26B - El dispositivo de acuerdo con el aspecto 13B, en donde el dispositivo comprende uno o más de una cámara, una computadora, un dispositivo móvil, un dispositivo receptor de transmisión o un decodificador.

[0306] Aspecto 27B - El dispositivo de acuerdo con el aspecto 13, en donde el dispositivo comprende una cámara configurada para capturar los datos de video.

[0307] Aspecto 1C - Un método para codificar o decodificar datos de video incluye determinar un vector de movimiento para un bloque de datos de video mediante el uso de un proceso de derivación del vector de movimiento del lado del decodificador que incluye llevar a cabo un proceso de búsqueda iterativa, en donde llevar a cabo el proceso de búsqueda iterativa incluye: determinar un costo mínimo a través de una iteración de búsqueda anterior; actualizar el costo mínimo general a través de una iteración de búsqueda actual; y terminar anticipadamente el proceso de búsqueda iterativa en función de una comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, y codificar o decodificar el bloque de datos de video mediante el uso del vector de movimiento.

[0308] Aspecto 2C - El método de acuerdo con el aspecto 1C, en donde la iteración de búsqueda anterior precede inmediatamente a la iteración de búsqueda actual en el proceso de búsqueda iterativa.

[0309] Aspecto 3C - El método de acuerdo con cualquiera de los aspectos 1C y 2C incluye además calcular el costo mínimo general a través de la iteración de búsqueda actual en función de al menos un subconjunto actual de posiciones candidatas.

[0310] Aspecto 4C - El método de acuerdo con cualquiera de los aspectos 1C a 3C, en donde terminar anticipadamente el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual comprende calcular una

diferencia entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual; y finalizar el proceso de búsqueda iterativa anticipadamente en función de la diferencia relativa a un umbral de diferencia.

[0311] Aspecto 5C - El método de acuerdo con el aspecto 4C, en donde el umbral de diferencia se basa en el tamaño del bloque.

[0312] Aspecto 6C - El método de acuerdo con cualquiera de los aspectos 1C a 5C, en donde terminar anticipadamente el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual comprende calcular una relación entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual; y finalizar el proceso de búsqueda iterativa anticipadamente en función de la relación relativa a un umbral de relación.

[0313] Aspecto 7C - El método de acuerdo con el aspecto 6C, en donde el umbral de relación es fijo o se basa en el tamaño del bloque.

[0314] Aspecto 8C - El método de acuerdo con cualquiera de los aspectos 1C a 7C, en donde terminar anticipadamente el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual comprende: calcular una diferencia entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual; calcular una relación entre costo mínimo a través de la iteración de búsqueda anterior y la diferencia entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual; y finalizar el proceso de búsqueda iterativa anticipadamente en función de la relación relativa a un umbral de relación.

[0315] Aspecto 9C - El método de acuerdo con el aspecto 8C, en donde el umbral de relación es fijo o se basa en el tamaño del bloque.

[0316] Aspecto 10C - El método de acuerdo con cualquiera de los aspectos 1C a 9C, en donde terminar anticipadamente el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual comprende: terminar el proceso de búsqueda iterativa anticipadamente en función de la comparación entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual y una comparación entre el costo mínimo a

través de la iteración de búsqueda anterior y un costo mínimo de la iteración de búsqueda actual.

[0317] Aspecto 11C - El método de acuerdo con cualquiera de los aspectos 1C a 10C, en donde el proceso de búsqueda iterativa incluye más de dos iteraciones.

[0318] Aspecto 12C - El método de acuerdo con cualquiera de los aspectos 1C a 11C, en donde realizar el proceso de búsqueda iterativa incluye realizar una coincidencia bilateral o una comparación de plantillas.

[0319] Aspecto 13C - Un dispositivo para procesar datos de video incluye una memoria configurada para almacenar los datos de video; y una circuitería de procesamiento configurada para: determinar un vector de movimiento para un bloque de los datos de video mediante un proceso de derivación del vector de movimiento del lado del decodificador que incluye un proceso de búsqueda iterativa, en donde, para realizar el proceso de búsqueda iterativa, el circuito de procesamiento está además configurado para: determinar un costo mínimo a través de una iteración de búsqueda anterior; actualizar el costo mínimo general a través de una iteración de búsqueda actual; y terminar anticipadamente el proceso de búsqueda iterativa en función de una comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, y codificar o decodificar el bloque de datos de video mediante el uso del vector de movimiento.

[0320] Aspecto 14C - El dispositivo de acuerdo con el aspecto 13C, en donde la iteración de búsqueda anterior precede inmediatamente a la iteración de búsqueda actual.

[0321] Aspecto 15C - El dispositivo de acuerdo con cualquiera de los aspectos 13C y 14C, en donde, para realizar el proceso de búsqueda iterativa, el circuito de procesamiento está configurado además para calcular el costo mínimo general a través de la iteración de búsqueda actual en función de al menos un subconjunto actual de posiciones candidatas.

[0322] Aspecto 16C - El dispositivo de acuerdo con cualquiera de los aspectos 13C a 15C, en donde, para terminar anticipadamente el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, el circuito de procesamiento se configura, además, para: calcular una diferencia entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual; y finalizar el proceso de búsqueda iterativa anticipadamente en función de la diferencia relativa a un umbral de diferencia.

[0323] Aspecto 17C - El dispositivo de acuerdo con el aspecto 16C, en donde el umbral de diferencia se basa en el tamaño del bloque.

[0324] Aspecto 18C - El dispositivo de acuerdo con cualquiera de los aspectos 13C a 17C, en donde, para terminar anticipadamente el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, el circuito de procesamiento se configura, además, para: calcular una relación entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual; y finalizar el proceso de búsqueda iterativa anticipadamente en función de la relación relativa a un umbral de relación.

[0325] Aspecto 19C - El dispositivo de acuerdo con el aspecto 18C, en donde el umbral de relación es fijo o se basa en el tamaño del bloque.

[0326] Aspecto 20C - El dispositivo de acuerdo con cualquiera de los aspectos 13C a 19C, en donde para terminar anticipadamente el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, el circuito de procesamiento se configura además para: calcular una diferencia entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual; calcular una relación entre costo mínimo a través de la iteración de búsqueda anterior y la diferencia entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual; y finalizar el proceso de búsqueda iterativa anticipadamente en función de la relación relativa a un umbral de relación.

[0327] Aspecto 21C - El dispositivo de acuerdo con el aspecto 20C, en donde el umbral de relación es fijo o se basa en el tamaño del bloque.

[0328] Aspecto 22C - El dispositivo de acuerdo con cualquiera de los aspectos 13C a 21C, en donde para terminar anticipadamente el proceso de búsqueda iterativa en función de la comparación del costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual, el circuito de procesamiento se configura además para: terminar el proceso de búsqueda iterativa anticipadamente en función de la comparación entre el costo mínimo a través de la iteración de búsqueda anterior y el costo mínimo general a través de la iteración de búsqueda actual y una comparación entre el costo mínimo a través de la iteración de búsqueda anterior y un costo mínimo de la iteración de búsqueda actual.

[0329] Aspecto 23C - El dispositivo de acuerdo con cualquiera de los aspectos 13C a

22C, en donde el proceso de búsqueda iterativa incluye más de dos iteraciones.

[0330] Aspecto 24C - El dispositivo de acuerdo con cualquiera de los aspectos 13C a 23C, en donde, para realizar el proceso de búsqueda iterativa, el circuito de procesamiento se configura para realizar una coincidencia bilateral o una comparación de plantillas.

[0331] Aspecto 25C - El dispositivo de acuerdo con cualquiera de los aspectos 13C a 24C, que comprende además una pantalla configurada para mostrar los datos de video.

[0332] Aspecto 26C - El dispositivo de acuerdo con cualquiera de los aspectos 13C a 25C, en donde el dispositivo comprende uno o más de una cámara, una computadora, un dispositivo móvil, un dispositivo receptor de transmisión o un decodificador.

[0333] Aspecto 27C - El dispositivo de acuerdo con cualquiera de los aspectos 13C a 26C, en donde el dispositivo comprende una cámara configurada para capturar los datos de video.

[0334] Debe reconocerse que, según el ejemplo, ciertos actos o eventos de cualquiera de las técnicas descritas en la presente se pueden llevar a cabo en una secuencia diferente, se pueden agregar, fusionar o excluir por completo (por ejemplo, no todos los actos o eventos descritos son necesarios para la práctica de las técnicas). Además, en algunos ejemplos, los actos o eventos se pueden llevar a cabo concurrentemente, por ejemplo, a través de procesamiento multihilo, procesamiento interrumpido o procesadores múltiples, en lugar de secuencialmente.

[0335] En uno o más ejemplos, las funciones descritas pueden implementarse en *hardware*, *software*, *firmware* o cualquier combinación de estos. Si se implementan en *software*, las funciones pueden almacenarse o transmitirse como una o más instrucciones o código en un medio legible por computadora y pueden ejecutarse mediante una unidad de procesamiento de *hardware*. Los medios legibles por computadora pueden incluir medios de almacenamiento legibles por computadora, que corresponden a un medio tangible tal como un medio de almacenamiento de datos, o medios de comunicación, incluso cualquier medio que facilite la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por computadora generalmente pueden corresponder a (1) medios de almacenamiento legibles por computadora tangibles no transitorios o (2) un medio de comunicación tal como una onda de señal o portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se puede acceder mediante una o más computadoras o uno o más

procesadores para obtener instrucciones, códigos 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 computadora.

[0336] A modo de ejemplo, y no de limitación, tal medio de almacenamiento legible por computadora puede comprender RAM, ROM, EEPROM, CD-ROM u otros dispositivos de almacenamiento en disco óptico, de almacenamiento en disco magnético u otro almacenamiento magnético, o cualquier otro medio que pueda utilizarse para almacenar el código de programación deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante una computadora. Además, cualquier conexión se denomina de manera adecuada medio legible por computadora. Por ejemplo, si las instrucciones se transmiten desde un sitio web, servidor u otra fuente remota mediante un cable coaxial, cable de fibra óptica, par trenzado, línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarroja, radio y microondas, entonces el cable coaxial, cable de fibra óptica, par trenzado, DSL o tecnologías inalámbricas, tales como infrarroja, radio y microondas, están incluidos en la definición de medio. Sin embargo, cabe aclarar que los medios de almacenamiento legibles por computadora y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en su lugar, se refieren a medios de almacenamiento tangibles no transitorios. Como se usa en la presente, disco incluye disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disquete y disco Blu-ray, en donde los discos reproducen datos magnéticamente y datos ópticamente con láseres. También deberían incluirse combinaciones de los anteriores dentro del alcance de medios legibles por computadora.

[0337] Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más DSP, microprocesadores de propósito general, ASIC, FPGA u otra circuitería lógica integrada o separada equivalente. En consecuencia, las expresiones "procesador" y "circuitería de procesamiento", como se usa en la presente, pueden referirse a cualquiera de las estructuras anteriores o cualquier otra estructura adecuada para la aplicación de las técnicas descritas en la presente. Además, en algunos aspectos, la funcionalidad descrita en la presente puede proporcionarse dentro de los módulos de *hardware* y/o *software* dedicados configurados para codificar y decodificar, o incorporarse en un códec combinado. Además, las técnicas podrían implementarse totalmente en uno o más circuitos o elementos lógicos.

[0338] Las técnicas de esta divulgación se pueden implementar en una amplia

variedad de dispositivos o aparatos, que incluyen un microteléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación, se describen distintos componentes, módulos o unidades para enfatizar los aspectos funcionales de los dispositivos configurados para realizar las técnicas divulgadas, pero no es necesario que sean realizadas por diferentes unidades de *hardware*. Más bien, como se describió anteriormente, diversas unidades pueden combinarse en una unidad de *hardware* de códec o proporcionarse mediante una colección de unidades de *hardware* interoperativas, que incluye uno o más procesadores como se describió anteriormente, junto con *software* y/o *firmware* adecuado.

[0339] Se han descrito diversos ejemplos. Estos y otros ejemplos se encuentran dentro del alcance de las siguientes reivindicaciones.