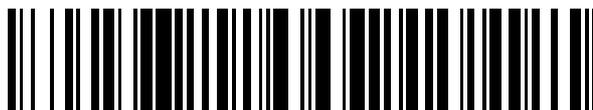


19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 492 923**

51 Int. Cl.:

**H04N 13/00** (2006.01)  
**H04N 19/597** (2014.01)  
**H04N 19/573** (2014.01)  
**H04N 19/70** (2014.01)  
**H04N 19/105** (2014.01)  
**H04N 19/172** (2014.01)  
**H04N 19/423** (2014.01)  
**H04N 19/103** (2014.01)  
**H04N 19/46** (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **15.10.2007 E 07826751 (5)**

97 Fecha y número de publicación de la concesión europea: **04.06.2014 EP 2087741**

54 Título: **Sistema y procedimiento para implementar una administración eficiente de memoria intermedia decodificada en codificación de video de vistas múltiples**

30 Prioridad:

**16.10.2006 US 852223 P**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**10.09.2014**

73 Titular/es:

**NOKIA CORPORATION (100.0%)  
Karakaari 7  
02610 Espoo, FI**

72 Inventor/es:

**CHEN, YING;  
WANG, YE-KUI y  
HANNUKSELA, MISKA**

74 Agente/Representante:

**VALLEJO LÓPEZ, Juan Pedro**

**ES 2 492 923 T3**

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

## DESCRIPCIÓN

Sistema y procedimiento para implementar una administración eficiente de memoria intermedia decodificada en codificación de video de vistas múltiples

5

**Campo de la invención**

La presente invención se refiere en general con la codificación de video. Más específicamente, la presente invención se refiere a la administración de la memoria intermedia de imágenes codificadas en la codificación de video de vistas múltiples.

10

**Antecedentes de la invención**

En la codificación de video de vistas múltiples, las secuencias de video producidas desde diferentes cámaras, cada una correspondiendo a diferentes vistas de una escena, son codificadas en un solo flujo de bits. Después de la decodificación, para mostrar cierta vista, las imágenes decodificadas que pertenecen a esa vista son reconstruidas y visualizadas. También es posible que más de una vista sea reconstruida y visualizada.

15

La codificación de video de vistas múltiples procesa una amplia variedad de aplicaciones, incluyendo video/televisión de punto de vista libre, TV tridimensional (3D) y aplicaciones de sondeo. Actualmente, el Equipo de Video Conjunto (JVT) de la Organización Internacional para la Estandarización (ISO)/Grupo de Expertos de Imágenes en Movimiento (MPEG) del Consorcio Internacional de Ingeniería (IEC) y el Grupo de Expertos en Codificación de Video de la Unión de Telecomunicación Internacional (ITU)-T está trabajando para desarrollar un estándar de codificación de video de vistas múltiples (MVC), el cual se está convirtiendo en una extensión del estándar ITU-T H.264, también conocido como ISO/IEC MPEG-4 Parte 10. Estos estándares borradores se mencionan en el presente documento como MVC y AVC, respectivamente. El último borrador del estándar MVC se describe en JVT-T208, "Joint Multiview Video Model (JMVM) 1.0", 20ª Reunión de la JVT, Klagenfurt, Austria, Julio de 2006, puede encontrarse en [ftp3.itu.ch/av-arch/ivt-site/2006/07\\_Klagenfurt/JVT-T208.zip](ftp3.itu.ch/av-arch/ivt-site/2006/07_Klagenfurt/JVT-T208.zip).

20

25

En JMVM 1.0, para cada grupo de imágenes (GOP), las imágenes de cualquier vista son contiguas en orden de decodificación. Esto se ilustra en la figura 1, donde la dirección horizontal indica el tiempo (siendo cada instante de tiempo representado por  $T_m$ ) y la dirección vertical indica la vista (estando cada vista representada por  $S_n$ ). Las imágenes de cada vista son agrupadas en GOPs, por ejemplo, las imágenes T1 a T8 en la figura 1 para cada vista forman un GOP. Esta disposición de orden de decodificación se llama como codificación de la primera vista. Debe indicarse que para las imágenes en una vista y en un GOP, a pesar de que su orden de decodificación es continuo sin ninguna otra imagen para insertarse entre cualquiera de las dos imágenes, su orden de decodificación puede cambiar internamente.

30

35

También es posible tener un orden de decodificación diferente al descrito para la codificación de la primera vista. Por ejemplo, las imágenes pueden disponerse de tal manera que las imágenes de cualquier ubicación temporal sean contiguas en el orden de decodificación. Esta disposición se muestra en la figura 2. Esta disposición de orden de decodificación se denomina codificación de primero el tiempo. También debe indicarse que el orden de decodificación de las unidades de acceso puede no ser idéntico al orden temporal.

40

Una estructura de predicción típica (que incluye predicción entre imágenes en cada vista y predicción entre vistas) para la codificación de video de vistas múltiples se muestra en la figura 2, donde las predicciones se indican mediante flechas, y el objeto apuntado-a utiliza el objeto apuntado-desde para referencia de predicción. La predicción entre imágenes dentro de una vista también se denomina predicción temporal, predicción entre vistas, o simplemente, interpredicción.

45

50

Una imagen de Actualización Instantánea de Decodificación (IDR) es una imagen intracodificada que hace que el proceso de decodificación marque todas las imágenes de referencia como "no usadas para referencia" inmediatamente después de decodificar la imagen IDR. Después de decodificar una imagen IDR, todas las siguientes imágenes codificadas en el orden de decodificación pueden decodificarse sin interpredicción de cualquier imagen decodificada antes que la imagen IDR.

55

En AVC y MVC, los parámetros de codificación que se mantienen sin cambios a través de una secuencia de video codificada se incluyen en un conjunto de parámetros de secuencias. Además de los parámetros que son esenciales para el proceso de decodificación, el conjunto de parámetros de secuencias puede contener opcionalmente información de utilización de video (VUI), la cual incluye parámetros que son importantes para almacenamiento temporal, sincronización de la salida de imágenes, representación, y reserva de recursos. Existen dos estructuras especificadas para llevar conjuntos de parámetros de secuencias - la unidad NAL del conjunto de parámetros de secuencias que contiene todos los datos para imágenes AVC en la secuencia, y la extensión del conjunto de parámetros de secuencias para MVC. Un conjunto de parámetros de imagen contiene los parámetros que probablemente no cambiarán en varias imágenes codificadas. Frecuentemente, el cambio de datos del nivel de imagen se repite en cada encabezado de líneas, y los conjuntos de parámetros de imágenes llevan los parámetros

60

65

de nivel de imagen restantes. La sintaxis H.264/AVC permite muchas instancias de conjuntos de parámetros de secuencias e imágenes, y cada instancia está identificada con un identificador único. Cada encabezado de línea incluye el identificador del conjunto de parámetros de imágenes que está activo para la decodificación de la imagen que contiene la línea, y cada conjunto de parámetros de imágenes contiene el identificador del conjunto de parámetros de secuencias activo. En consecuencia, la transmisión de los conjuntos de parámetros de imágenes y secuencias no tiene que sincronizarse con precisión con la transmisión de las líneas. Más Bien, es suficiente que los conjuntos de parámetros de secuencias e imágenes activos sean recibidos en cualquier momento antes de que sean referenciados, lo cual permite la transmisión de conjuntos de parámetros usando un mecanismo de transmisión más fiable comparado con los protocolos usados para los datos de líneas. Por ejemplo, conjuntos de parámetros pueden incluirse como un parámetro MIME en la descripción de la sesión para las sesiones del Protocolo en Tiempo Real (RTP) de H.264/AVC. Se recomienda usar un mecanismo de transmisión fiable fuera de banda siempre que sea posible en la aplicación en uso. Si se transmiten conjuntos de parámetros dentro de la banda, estos pueden repetirse para mejorar la robustez de errores.

Como se describe en el presente documento, una imagen ancla es una imagen codificada en la cual todas las líneas solo hacen referencia a líneas con el mismo índice temporal, es decir, solo líneas en otras vistas y no líneas en imágenes anteriores de la vista actual. Una imagen ancla se señala estableciendo un `anchor_pic_flag` en 1. Después de decodificar la imagen ancla, todas las imágenes codificadas posteriores en el orden de visualización son capaces de decodificarse sin interpretación de ninguna imagen decodificada antes de la imagen ancla. Si una imagen en una vista es una imagen ancla, entonces todas las imágenes con el mismo índice temporal en otras vistas son también imágenes ancla. En consecuencia, la decodificación de cualquier vista puede iniciarse desde un índice temporal que corresponde a imágenes ancla.

La sincronización de la salida de imágenes, tal como registro de tiempo de salida, no se incluye en la parte integral de flujos de bits de AVC o MVC. Sin embargo, un valor de conteo de orden de imágenes (POC) se deriva para cada imagen y es no decreciente con el incremento en la posición de la imagen en el orden de salida en relación con la imagen IDR previa o una imagen que contiene una operación de control de administración de memoria que marca todas las imágenes como "no usadas para referencia". Por lo tanto, el POC indica el orden de salida de las imágenes. También se usa en el proceso de decodificación para escalar implícitamente vectores de movimiento en los modos directos de líneas doblemente predictivas, para pesos derivados implícitamente en la predicción ponderada, y para la inicialización de una lista de imágenes de referencia de líneas B. Adicionalmente, el POC se utiliza también en la verificación de la conformidad del orden de salida.

Los valores de POC pueden codificarse con uno de los tres modos serializados en el conjunto de parámetros de secuencias activos. En el primer modo, se incluye el número seleccionado de los bits menos significativos del valor de POC en cada encabezado de línea. En el segundo modo, los incrementos relativos de POC como función de la posición de la imagen en el orden de decodificación en la secuencia de video codificada se codifican en el conjunto de parámetros de secuencias. Además, las desviaciones del valor de POC derivado del conjunto de parámetros de secuencias pueden indicarse en los encabezados de líneas. En el tercer modo, el valor de POC se deriva del orden de decodificación suponiendo que la decodificación y el orden de salida son idénticos. Además, solo una imagen de no referencia puede aparecer consecutivamente cuando se utiliza el tercer modo.

`nal_ref_idc` es un elemento de sintaxis de 2 bits en el encabezado de la unidad NAL. El valor de `nal_ref_idc` indica la relevancia de la unidad NAL para la reconstrucción de valores de muestra. Los valores diferentes de cero de `nal_ref_idc` deben usarse para unidades NAL de partición de datos de línea y línea codificada de las imágenes de referencia, así como para unidades NAL del conjunto de parámetros. El valor `nal_ref_idc` debe ser igual a 0 para líneas y particiones de datos de líneas de imágenes de no referencia y para unidades NAL que no afectan la reconstrucción de valores de muestras, tales como unidades NAL de información de mejora suplementaria. En el diseño de alto nivel de H.264/AVC, se permitió que las especificaciones externas (es decir, cualquier sistema o especificación que utilice o que haga referencia a H.264/AVC) especificaran una interpretación de los valores diferentes de cero de `nal_ref_idc`. Por ejemplo, el formato de carga útil RTP para H.264/AVC, Solicitud de Comentarios (RFC) 3984 (que puede encontrarse en [www.ietf.org/rfc/rfc3984.txt](http://www.ietf.org/rfc/rfc3984.txt) y que se incorpora aquí como referencia) especificó fuertes recomendaciones acerca del uso de `nal_ref_idc`. En otras palabras, algunos sistemas han establecido prácticas para fijar e interpretar los valores diferentes de cero de `nal_ref_idc`. Por ejemplo, un mezclador de RTP podría establecer `nal_ref_idc` de acuerdo con el tipo de unidad NAL, por ejemplo, `nal_ref_idc` se fija en 3 para unidades IDR NAL. Como MVC es una extensión compatible hacia atrás del estándar H.264/AVC, es deseable que los elementos del sistema de conocimiento de H.264/AVC sean también capaces de manipular flujos de MVC. Por lo tanto, es indeseable que la semántica de valor diferente de cero particular de `nal_ref_idc` se especifique de manera diferente en la especificación de MVC en comparación con cualquier valor diferente de cero de `nal_ref_idc`.

Las imágenes decodificadas usadas para predecir imágenes codificadas posteriores y para la salida futura son almacenadas temporalmente en una memoria intermedia de imágenes decodificadas (DPB). Para utilizar eficientemente la una memoria intermedia, el proceso de administración de DPB, incluyendo el proceso de almacenamiento de imágenes decodificadas dentro del DPB, el proceso de marcación de imágenes de referencia, los procesos de salida y de retirada de imágenes decodificadas del DPB, deben especificarse.

El proceso para marcación de imágenes de referencia en AVC es generalmente de la siguiente manera. El número máximo de imágenes de referencia utilizadas para interpredicción, denominado M, se indica en el conjunto de parámetros de secuencias activas. Cuando se decodifica una imagen, esta es marcada como "usada para referencia". Si la decodificación de la imagen de referencia ocasiona que más de M imágenes sean marcadas como "usadas para referencia", entonces por lo menos una imagen debe marcarse como "no usada para referencia". El proceso de retirada de DPB retiraría entonces imágenes marcadas como "no usadas para referencia" del DPB si tampoco necesitan ser retiradas.

Existen dos tipos de operaciones para la marcación de imágenes de referencia: control de memoria adaptativa y ventana deslizante. El modo de operación para la marcación de imágenes de referencia se selecciona con base en la imagen. El control de imagen adaptativa requiere la presencia de comandos de operación de control de administración de memoria (MMCO) en el flujo de bits. Las operaciones de control de administración de memoria permiten la señalización explícita de qué imágenes están marcadas como "no usadas para referencia", la asignación de índices de largo plazo a imágenes de referencia de corto plazo, el almacenamiento de imágenes actuales como imágenes de largo plazo, el cambio de una imagen de corto plazo a la imagen de largo plazo, y la asignación del índice de largo plazo máximo permitido (MaxLongTermFrameIdx) para imágenes de largo plazo. Si el modo de operación de ventana deslizante esta en uso y hay M imágenes marcadas como "usadas para referencia", entonces la imagen de referencia de corto plazo que fue la primera imagen decodificada entre las imágenes de referencia de corto plazo que se marcaron como "usadas para referencia" se marca como "no usada para referencia". En otras palabras, el modo de operación de ventana deslizante resulta en una operación de almacenamiento en memoria intermedia de primero en entrar/primerero en salir entre imágenes de referencia de corto plazo.

Cada imagen de corto plazo está asociada con una variable PicNum que se deriva del elemento de sintaxis frame\_num. Cada imagen de largo plazo está asociada con un LongTermPicNum variable que se deriva del elemento de sintaxis long\_term\_frame\_idx, el cual se señala por el comando MMCO. PicNum se deriva del elemento de sintaxis FrameNumWrap, dependiendo de si el cuadro o campo está codificado o decodificado. Para cuadros en los que PicNum es igual a FrameNumWrap, FrameNumWrap se deriva de FrameNum, y FrameNum se deriva directamente de frame\_num. Por ejemplo, en la codificación de cuadros de AVC, a FrameNum se le asigna el mismo valor que frame\_num, y FrameNumWrap se define como sigue:

```

if( FrameNum > frame_num )
  FrameNumWrap = FrameNum - MaxFrameNum
else
  FrameNumWrap = FrameNum

```

LongTermPicNum se deriva del índice de cuadro de largo plazo (LongTermFrameIdx) asignado para la imagen. Para cuadros, LongTermPicNum es igual a LongTermFrameIdx. Frame\_num es un elemento de sintaxis en cada encabezado de línea. El valor de frame\_num para un cuadro o un par de campos complementarios aumenta esencialmente en uno, en módulo aritmético, en relación con el frame\_num del cuadro de referencia anterior o el par de campos complementarios de referencia. En imágenes IDR, el valor de frame\_num es cero. Para imágenes que contienen una operación de control de administración de memoria que marca todas las imágenes como "no usadas para referencia", el valor de frame\_num se considera que es cero después de la decodificación de la imagen.

Los comandos de MMCO utilizan PicNum y LongTermPicNum para indicar la imagen objetivo para el comando de la siguiente manera. Para marcar una imagen de corto plazo como "no usada para referencia", la diferencia de PicNum entre la imagen actual p y la imagen de destino r se señala en el comando de MMCO. Para marcar una imagen de largo plazo como "no usada para referencia", el LongTermPicNum de la imagen a retirar r se señala en el comando de MMCO. Para almacenar la imagen actual p como una imagen de largo plazo, se señala un long\_term\_frame\_idx con el comando de MMCO. Este índice está asignado a la imagen de largo plazo recientemente almacenada como el valor de LongTermPicNum. Para cambiar una imagen r de ser una imagen de corto plazo a una imagen de largo plazo, se señala una diferencia PicNum entre la imagen actual p y la imagen r en el comando de MMCO, el longterm\_frame\_idx se señala en el comando de MMCO, y el índice es asignado a esta imagen de largo plazo.

Cuando se pueden usar imágenes de referencias múltiples, cada imagen de referencia debe ser identificada. En AVC, la identificación de una imagen de referencia utilizada para un bloque codificado es como sigue. En primer lugar, todas las imágenes de referencia almacenadas en el DPB para referencia de predicción de imágenes futuras se marcan como "usadas para referencia de corto plazo" (imágenes de corto plazo) o "usadas como referencia de largo plazo" (imágenes de largo plazo). Cuando se decodifica una línea codificada, se construye una lista de imágenes de referencia. Si la línea codificada es una línea doblemente predicha, entonces también se construye una segunda lista de imágenes de referencia. Una imagen de referencia usada para un bloque codificado es identificada entonces por el índice de la imagen de referencia utilizada en la lista de imágenes de referencia. El índice se codifica en el flujo de bits cuando se usa más de una imagen de referencia.

El proceso de construcción de la lista de imágenes de referencia es el siguiente. Para simplicidad, se supone que solo se necesita una lista de imágenes de referencia. Primero, se construye una lista de imágenes de referencia

inicial que incluye todas las imágenes de corto plazo y de largo plazo. Se realiza entonces el reordenamiento de la lista de imágenes de referencia (RPLR) cuando el encabezado de línea contiene comandos de RPLR. El proceso de RPLR puede reordenar las imágenes de referencia en un orden diferente al orden en la lista inicial. Finalmente, la lista final se construye manteniendo solo un número de imágenes al principio de la lista posiblemente reordenada, indicándose el número mediante otro elemento de sintaxis en el encabezado de línea o el conjunto de parámetros de imágenes referido por la línea.

Durante el proceso de inicialización, todas las imágenes de corto plazo y de largo plazo se consideran como candidatos para listas de imágenes de referencia para la imagen actual. Independientemente de si la imagen actual es una imagen B o P, las imágenes de largo plazo se colocan después de las imágenes de corto plazo en RefPicList0 (y RefPicList1 disponible para líneas B). Para imágenes P, la lista de imágenes de referencia inicial para RefPicList0 contiene todas las imágenes de referencia de corto plazo ordenadas en orden descendente de PicNum. Para imágenes B, aquellas imágenes de referencia obtenidas de todas las imágenes de corto plazo están ordenadas mediante una regla relacionada con el número de POC actual y el número de la imagen de referencia para RefPicList0, las imágenes de referencia con POC más pequeño (en comparación con el POC actual) se consideran primero y se insertan en la RefPicList0 con el orden descendente de POC. Después las imágenes con POC más grande son anexadas con el orden ascendente de POC. Para RefPicList1 (si está disponible), las imágenes de referencia con POC más grande (en comparación con el POC actual) se consideran primero y se insertan en la RefPicList1 con orden ascendente de POC. Las imágenes con POC más pequeño se anexan entonces en orden descendente del POC. Después de considerar todas las imágenes de referencia de corto plazo, las imágenes de referencia de largo plazo se anexan en orden ascendente de LongTermPicNum, tanto para imágenes P como para B.

El proceso de reordenamiento es invocado por medio de comandos de RPLR, los cuales incluyen cuatro tipos. El primer tipo es un comando para especificar una imagen de corto plazo con PicNum más pequeño (en comparación con un PicNum temporalmente predicho) que será movida. El segundo tipo es un comando para especificar una imagen de corto plazo con PicNum más grande que será movida. El tercer tipo es un comando para especificar una imagen de largo plazo con cierto LongTermPicNum que será movida y el final del ciclo de RPLR. Si la imagen actual es doblemente predicha, entonces hay dos ciclos, uno para una lista de referencia hacia adelante y la otra para una lista de referencia hacia atrás.

El PicNum predicho denominado picNumLXPred se inicializa como el PicNum de la imagen actual codificada. Esto se establece en el PicNum de la imagen que acaba de moverse después de cada proceso de reordenamiento para una imagen de corto plazo. La diferencia entre el PicNum de la imagen actual que es reordenada y picNumLXPred se señalará en el comando de RPLR. La imagen indicada a reordenar se mueve al principio de la lista de imágenes de referencia. Después de que se complete el proceso de reordenamiento, toda una lista de imágenes de referencia se truncará con base en el tamaño de la lista de imágenes de referencia activa, la cual es  $\text{num\_ref\_idx\_IX\_active\_minus1}+1$  (X es igual a 0 ó 1 corresponde a RefPicList0 y RefPicList1 respectivamente).

El decodificador de referencia hipotético (HRD), especificado en el Anexo C del estándar H.264/AVC, se usa para verificar la conformidad del flujo de bits y del decodificador. El HRD contiene una memoria intermedia de imágenes codificadas (CPB), un proceso de decodificación instantánea, una memoria intermedia de imágenes decodificadas (DPB), y un bloque de recortes de imágenes de salida. La CPB y el proceso de decodificación instantáneo se especifican de manera similar a cualquier otro estándar de codificación de video, y el bloque de recorte de imágenes de salida simplemente recorta esas muestras de la imagen decodificada que están fuera de los alcances de las imágenes de salida señalizadas. El DPB se introdujo en H.264/AVC con el fin de controlar los recursos de memoria requeridos para decodificar flujos de bits de conformidad.

Existen dos razones para almacenar temporalmente imágenes decodificadas, para referencias en interpredicción y para reordenar imágenes decodificadas en el orden de salida. Como el estándar H.264/AVC proporciona una gran flexibilidad tanto para marcación de imágenes de referencia como de reordenamiento de salida, las memorias intermedias separadas para almacenamiento temporal de imágenes de referencia y almacenamiento temporal de imágenes de salida podrían ser un desperdicio de recursos de memoria. Por lo tanto, el DPB incluye un proceso de almacenamiento temporal de imágenes decodificadas unificado para imágenes de referencia y reordenamiento de salida. Una imagen decodificada es retirada del DPB cuando ya no se utiliza como referencia ni se necesita para salida. El tamaño máximo del DPB que se permite usar para flujos de bits se especifica en las definiciones de Nivel (Anexo A) del estándar H.264/AVC.

Existen dos tipos de conformidad para decodificadores: conformidad de sincronización de salida y conformidad del orden de salida. Para conformidad de sincronización de salida, un decodificador debe sacar imágenes en tiempos idénticos en comparación con HRD. Para la conformidad del orden de salida, sólo se toma en cuenta el orden correcto de la imagen de salida. Se supone que el DPB del orden de salida contiene un número máximo permitido de memorias intermedias de cuadros. Un cuadro se retira del DPB cuando ya no se utiliza como referencia ni se necesita para la salida. Cuando el DPB se llena, el cuadro más anterior en el orden de salida se emite hasta que se desocupa por lo menos una memoria intermedia de cuadros.

La escalabilidad temporal se realiza mediante la estructura jerárquica de GOP de imágenes B usando sólo herramientas AVC. Un GOP de escalabilidad temporal típico usualmente incluye una imagen clave, la cual es codificada como un cuadro I o P, y otras imágenes que son codificadas como imágenes B. Esas imágenes B son codificadas jerárquicamente con base en el POC. La codificación de un GOP sólo necesita las imágenes clave del GOP previo además de esas imágenes en el GOP. El número relativo de POC (POC menos el POC de imagen ancla anterior) se denomina POCIdInGOP en la implementación. Cada POCIdInGOP puede tener una forma de  $POCIdInGOP = 2^x y$  (donde y es un número impar). Las imágenes con el mismo valor de x pertenecen al mismo nivel temporal, el cual se indica como L-x (donde  $L = \log_2(GOP\_longitud)$ ). Sólo las imágenes con el nivel temporal más alto L no se almacenan como imágenes de referencia. Normalmente, las imágenes en un nivel temporal sólo pueden usar imágenes en niveles temporales inferiores como referencias para soportar la escalabilidad temporal, es decir, pueden depositarse imágenes de nivel temporal más alto sin afectar la decodificación de las imágenes de nivel temporal más bajo. Similarmente, la misma estructura jerárquica puede aplicarse en la dimensión de vistas para la escalabilidad de vistas.

En el JMVM actual, frame\_num se codifica por separado y se señala para cada vista, es decir, el valor de frame\_num se incrementa en relación con el cuadro de referencia previo o el par de campos complementarios de referencia dentro de la misma vista que la imagen actual. Además, las imágenes en todas las vistas comparten la misma memoria intermedia DPB. Para manipular globalmente la construcción de listas de imágenes de referencia y la administración de imágenes de referencia, la generación de FrameNum y POC se redefinen de la siguiente manera:

```
FrameNum = frame_num *(1 + num_views_minus_1) + view_id
PicOrderCnt() = PicOrderCnt() * (1+num_views_minus_1) + view_id;
```

JMVM sigue básicamente la misma marcación de imágenes de referencia que la utilizada para AVC. La única diferencia es que, en JMVM el FramNum se redefine y que FrameNumWrap se redefine de la siguiente manera:

```
if (FrameNum > frame num * (1 + num_views_minus_1) + view_id)
FrameNumWrap = FrameNum - MaxFrameNum * (1 + num_views_minus_1) + view_id
else
FrameNumWrap = FrameNum
```

En el estándar JMVM actual, las imágenes de referencia entre vistas se especifican implícitamente en la extensión de SPS (Conjunto de Parámetros de Secuencias), donde se especifican el número activo de listas de referencias entre vistas y la identificación de vistas de esas imágenes. Esta información es compartida por todas las imágenes que hacen referencia al mismo SPS. El proceso de construcción de la lista de imágenes de referencia primero realiza la inicialización de la lista de imágenes de referencia, reordenando y truncando en la misma forma que en AVC, pero tomando en cuenta todas las imágenes de referencia almacenadas en el DPB. Las imágenes con identificaciones de vistas especificadas en el SPS y dentro del mismo eje temporal (es decir, que tienen el mismo tiempo de captura/salida) son anexadas entonces a la lista de referencias en el orden en el que están listadas en el SPS.

Desafortunadamente, los diseños de JSVM anteriores dan lugar a varios problemas. Primero, algunas veces es deseable que se produzca el cambio de vistas decodificadas (por medio de un decodificador), transmitidas (por medio de un emisor) o enviadas (por medio de una pasarela o MANE) en un índice de tiempo diferente al que corresponde a las imágenes ancla. Por ejemplo, una vista base puede comprimirse para la eficiencia de codificación más alta (se usa bastante la predicción temporal) y las imágenes ancla se codifican infrecuentemente. En consecuencia, las imágenes ancla para otras vistas también se producen infrecuentemente, debido a que se sincronizan a través de todas las vistas. La sintaxis de JMVM actual no incluye la señalización de una imagen desde la cual pueda iniciarse la decodificación de cierta vista (a menos que todas las vistas del índice de tiempo contengan una imagen ancla).

En segundo lugar, las vistas de referencia permitidas para la predicción entre vistas se especifican para cada vista (y por separado para imágenes ancla y no ancla). Sin embargo, dependiendo de la similitud entre una imagen que es codificada y una imagen potencial en el mismo eje temporal y en una vista de referencia potencial, la predicción entre vistas puede o no realizarse en el codificador. El estándar JMVM actual utiliza nal\_ref\_idc para indicar si una imagen se usa para predicción intra-vistas o inter-vistas, pero no puede indicar por separado si una imagen se usa para predicción intra-vistas y/o predicción inter-vistas. Además, de acuerdo con JMVM 1.0, para la vista compatible de AVC, nal\_ref\_idc debe establecerse no igual a 0, incluso si la imagen no se utiliza para predicción temporal cuando se usa sólo para referencia de predicción inter-vistas. Consecuentemente, si sólo esa vista se decodifica y se retira, se necesita un tamaño de DPB adicional para el almacenamiento de tales imágenes cuando esas imágenes puedan retirarse tan pronto como sean decodificadas.

En tercer lugar, se aprecia que el proceso de marcación de imágenes de referencia especificado en JMVM 1.0 es básicamente idéntico al del proceso AVC, excepto por la redefinición de FramNum, FrameNumWrap y consecuentemente PicNum. Por lo tanto, surgen varios problemas especiales. Por ejemplo, este proceso no puede

manejar eficientemente la administración de imágenes decodificadas que se requieren para almacenar temporalmente para la predicción inter-vistas, particularmente cuando esas imágenes no se utilizan para referencia de predicción temporal. La razón es que el proceso de administración de DPB especificado en el estándar AVC estaba dirigido a codificación de una sola vista. En la codificación de una sola vista tal como en el estándar AVC, las imágenes decodificadas que necesitan almacenarse temporalmente para la referencia de predicción temporal o salida futura pueden retirarse de la memoria intermedia cuando ya no se necesitan para referencia de predicción temporal y salida futura. Para permitir la retirada de una imagen de referencia tan pronto como ya no se necesite para referencia de predicción temporal y salida futura, el proceso de marcación de imágenes de referencia se especifica de tal manera que puede conocerse inmediatamente después de que una imagen de referencia ya no se necesita para referencia de predicción temporal. Sin embargo, cuando se trata de imágenes para referencia de predicción inter-vistas, hay una carencia de una forma de saber inmediatamente después de que una imagen ya no es necesaria para referencia de predicción inter-vistas. Consecuentemente, las imágenes para referencia de predicción inter-vistas pueden ser almacenadas temporalmente de manera innecesaria en el DPB, lo cual reduce la eficiencia del uso de la memoria intermedia.

En otro ejemplo, dada la forma para recalcular el PicNum, si el modo de operación de la ventana deslizante está en uso y el número de imágenes de corto plazo y de largo plazo es igual al máximo, la imagen de referencia de corto plazo que tiene el FrameNumWrap más pequeño se marca como "no usado como referencia". Sin embargo, debido al hecho de que esta imagen no es necesariamente la imagen codificada más anterior debido a que el orden del FrameNum en el JMVM actual no sigue el orden de decodificación, la imagen de referencia de ventana deslizante no opera de manera óptima en el JMVM actual. Adicionalmente, debido al hecho de que el PicNum se deriva del FrameNumWrap redefinido y escalado, la diferencia entre los valores de PicNum de dos imágenes codificadas se escalaría en promedio. Por ejemplo, es de ayuda suponer que existen dos imágenes en la misma vista y que tienen un frame\_num igual a 3 y 5, respectivamente. Cuando hay sólo una vista, es decir, el flujo de bits es un flujo AVC, entonces la diferencia de los dos valores de PicNum sería 2. Cuando se codifica la imagen que tiene un frame\_num igual a 5, si se necesita un comando de MMCO para marcar la imagen que tiene PicNum igual a 3 como "no usada para referencia", entonces la diferencia de los dos valores menos 1 es igual a 1, lo cual será señalizado en el MMCO. Este valor necesita 3 bits. Sin embargo, si hay 256 vistas, entonces la diferencia de los dos valores de PicNum menos 1 se convertiría en 511. En este caso, se requieren 19 bits para la señalización del valor. Consecuentemente, los comandos de MMCO son codificados en forma mucho menos eficiente. Normalmente, el mayor número de bits es igual a  $2 \cdot \log_2$  (número de vistas) para un comando de MMCO del JMVM actual en comparación con la codificación de vista simple de H.264/AVC.

Un cuarto conjunto de problemas tiene que ver con el proceso de construcción de la lista de imágenes de referencia especificado en JMVM 1.0. El proceso de inicialización de listas de imágenes de referencia considera a las imágenes de referencia de todas las vistas antes del proceso de reordenamiento. Sin embargo, debido al hecho de que las imágenes de las otras vistas usadas para predicción inter-vistas son anexadas a la lista después de trunca la lista, las imágenes de referencia de las otras vistas no aparecen en la lista de imágenes de referencia después de reordenar y trunca en cualquier forma. Por lo tanto, no se necesita la consideración de esas imágenes en el proceso de inicialización. Además, pueden aparecer imágenes de referencia ilegales (imágenes que tienen una view\_id diferente de la imagen actual y no están alineadas temporalmente con la imagen actual) y pueden aparecer imágenes de referencia inter-vistas repetidas en la lista de imágenes de referencia finalmente construida.

El proceso de inicialización de la lista de imágenes de referencia opera como se lista en las siguientes etapas: (1) Todas las imágenes de referencia se incluyen en la lista inicial independientemente de su view\_id y de si están temporalmente alineadas con la imagen actual. En otras palabras, la lista de imágenes de referencia inicial puede contener imágenes de referencia ilegales (imágenes que tienen una view\_id diferente que la imagen actual y no están temporalmente alineadas con la imagen actual). Sin embargo, en la codificación de vistas primero, el comienzo de la lista inicial contiene imágenes de referencia de la misma vista que la imagen actual. (2) Pueden reordenarse tanto imágenes de referencia intra-vistas como imágenes inter-vistas. Después de reordenar, el principio de la lista puede contener aún imágenes de referencia ilegales. (3) La lista está truncada, pero la lista truncada puede contener aún imágenes de referencia ilegales. (4) Las imágenes de referencia inter-vistas son anexadas a la lista en el orden en el que aparecen en la extensión de MVC de SPS.

Adicionalmente, el proceso de reordenamiento de la lista de imágenes de referencia especificado en JMVM 1.0 no permite el reordenamiento de cuadros inter-vistas, los cuales siempre se ponen al final de la lista en el orden en el que aparecen en la extensión de MVC de SPS. Esto ocasiona menos flexibilidad para la construcción de la lista de imágenes de referencia, lo cual resulta en una reducida eficiencia de compresión, cuando el orden por omisión de los cuadros de referencia inter-vistas no es óptimo o ciertos cuadros de referencia inter-vistas tienen más probabilidad de ser utilizados para predicción que ciertos cuadros de referencia intra-vistas. Adicionalmente, similar a los comandos MMCO, debido al hecho de que PicNum se deriva de FrameNumWrap redefinido y escalado, ya no se requieren las palabras de código de VLC para codificar comandos RPLR que incluyen la señalización de una diferencia entre los valores de PicNum menos 1 en comparación con la codificación de vista simple del estándar H.264/AVC.

El documento PURVIN PANDIT ET AL. "MVC high-level syntax for random access", ITU STUDY GROUP 16 –

VIDEO CODING EXPERTS GROUP – ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 E ITU-T SG16 Q6), XX, XX, nº M13715, 12 de julio de 2006 (12-07-2006), divulga una sintaxis de alto nivel en el marco de codificación de video de múltiples vistas. Este documento define una imagen ancla como una imagen que sólo permite la predicción a partir de otras vistas cuando la señal "nal\_ref\_idc" en el encabezado de unidad NAL es igual a 3 para facilitar el acceso aleatorio al decodificador.

### Sumario de la invención

La presente invención proporciona un sistema y un procedimiento mejorado para implementar la administración eficiente de la memoria intermedia de imágenes decodificadas en la codificación de video de vistas múltiples. En un ejemplo, se utiliza una nueva señal para indicar si la decodificación de una vista puede iniciarse desde cierta imagen. En un ejemplo más particular, esta señal se señala en el encabezado de la unidad NAL. En una realización, se utiliza una nueva señal para indicar si una imagen se usa para referencia de predicción inter-vistas, mientras que el elemento de sintaxis nal\_ref\_idc sólo indica si una imagen se utiliza para referencia de predicción temporal. Esta señal también puede señalizarse en el encabezado de la unidad NAL. En otro ejemplo, se utiliza un conjunto de procedimientos de marcación de imágenes de referencia para administrar eficientemente las imágenes decodificadas. Estos procedimientos pueden incluir tanto una ventana deslizante como mecanismos de control de memoria adaptativa. En otro ejemplo, se utiliza un conjunto de nuevos procedimientos de construcción de listas de imágenes de referencia e incluyen tanto la inicialización como el reordenamiento de listas de imágenes de referencia.

Estas y otras ventajas y características de la invención, junto con la organización y la manera de operación de la misma, serán evidentes a partir de la siguiente descripción detallada cuando se considere junto con los dibujos adjuntos, donde los elementos semejantes tienen números semejantes a lo largo de los varios dibujos descritos a continuación.

### Breve descripción de los dibujos

La figura 1 es una disposición de imágenes en una disposición de codificación de la vista primero;  
 La figura 2 es una disposición de imágenes de una disposición de codificación de primero el tiempo;  
 La figura 3 es una ilustración de un ejemplo de una estructura de predicción temporal e inter-vistas de MVC;  
 La figura 4 es un diagrama general de un sistema en el cual puede implementarse la presente invención;  
 La figura 5 es una vista en perspectiva de un dispositivo móvil que puede usarse en la implementación de la presente invención; y  
 La figura 6 es una representación esquemática de los circuitos del dispositivo móvil de la figura 5.

### Descripción detallada de la invención

La figura 4 muestra un sistema de comunicaciones multimedia genérico para usarse con la presente invención. Como se muestra en la figura 4, una fuente de datos 100 proporciona una señal fuente en un formato analógico, digital no comprimido, o digital comprimido, o cualquier combinación de estos formatos. Un codificador 110 codifica la señal fuente a un flujo de bits de medios codificados. El codificador 110 puede ser capaz de codificar más de un tipo de medio, tal como audio y video, o puede requerirse más de un codificador 110 para codificar diferentes tipos de medios de la señal fuente. El codificador 110 también puede obtener una entrada producida sintéticamente, tal como gráficos y texto, o puede ser capaz de producir flujos de bits codificados de medios sintéticos. A continuación, se considera sólo el procesamiento de un flujo de bits de medios codificados de un tipo de medio para simplificar la descripción. Deberá notarse, sin embargo, que los servicios de difusión en tiempo real comprenden normalmente varios flujos (normalmente por lo menos un flujo de audio, video y subtítulo de texto). También deberá notarse que el sistema puede incluir muchos codificadores, pero a continuación se considera sólo un codificador 110 para simplificar la descripción sin una falta de generalidad.

El flujo de bits de medios codificados es transferido a un almacenamiento 120. El almacenamiento 120 puede comprender cualquier tipo de memoria masiva para almacenar el flujo de bits de medios codificados. El formato del flujo de bits de medios codificados en el almacenamiento 120 puede ser un formato de flujo de bits autónomo elemental, o uno o más flujos de bits de medios codificados pueden estar encapsulados en un archivo contenedor. Algunos sistemas operan "en directo", es decir, omiten el almacenamiento y transfieren el flujo de bits de medios codificados del codificador 110 directamente al emisor 130. El flujo de bits de medios codificados es transferido entonces al emisor 130, también denominado servidor, con base en las necesidades. El formato usado en la transmisión puede ser un formato de flujo de bits autónomo elemental, un formato de flujo de paquetes, o uno o más flujos de bits de medios codificados pueden estar encapsulados en un archivo contenedor. El codificador 110, el almacenamiento 120, y el emisor 130 pueden residir en el mismo dispositivo físico o pueden estar incluidos en dispositivos separados. El codificador 110 y el emisor 130 pueden operar con contenido en tiempo real en directo, en cuyo caso el flujo de bits de medios codificados normalmente no se almacena permanentemente, sino que se almacena temporalmente por periodos pequeños de tiempo en el codificador de contenido 110 y/o en el emisor 130 para suavizar las variaciones en el retardo del procesamiento, en el retardo de transferencia, y en la tasa de bits de medios codificados.

El emisor 130 envía el flujo de bits de medios codificados usando una pila de protocolos de comunicación. La pila puede incluir pero no se limita a un Protocolo de Transporte en Tiempo Real (RTP), Protocolo de Datagramas de Usuario (UDP), y Protocolo de Internet (IP). Cuando la pila de protocolos de comunicación está orientada a paquetes, el emisor 130 encapsula el flujo de bits de medios codificados en paquetes. Por ejemplo, cuando se usa el RTP, el emisor 130 encapsula el flujo de bits de medios codificados en paquetes de RTP de acuerdo con un formato de carga útil de RTP. Normalmente, cada tipo de medio tiene un formato de carga útil de RTP dedicado. Deberá notarse nuevamente que un sistema puede contener más de un emisor (130), pero por simplicidad, la siguiente descripción sólo considera un emisor 130.

El emisor 130 puede o no estar conectado a una pasarela 140 a través de una red de comunicación. La pasarela 140 puede realizar diferentes tipos de funciones, tales como el traslado de un flujo de paquetes de acuerdo con una pila de protocolos de comunicación hacia otra pila de protocolos de comunicación, reuniendo y bifurcando los flujos de datos, y la manipulación de los flujos de datos de acuerdo con el enlace descendente y/o las capacidades receptoras, tal como el control de la tasa de bits del flujo enviado de acuerdo con las condiciones de red del enlace descendente prevaleciente. Ejemplos de pasarelas 140 incluyen unidades de control de conferencias multipunto (MCUs), pasarelas entre telefonía de video conmutada por circuito y conmutada por paquete, servidores de pulsar para hablar por celular (PoC), encapsuladores IP, en sistemas de difusión de video digital para dispositivos móviles (DVB-H), o decodificadores que envían transmisiones difundidas localmente a redes inalámbricas caseras. Cuando se utiliza RTP, la pasarela 140 se denomina mezclador RTP y actúa como punto final de una conexión RTP.

El sistema incluye uno o más receptores 150, normalmente capaces de recibir, demodular y desencapsular la señal transmitida a un flujo de bits de medios codificados. El flujo de bits de medios codificados normalmente es procesado adicionalmente por un decodificador 160, cuya salida es uno o más flujos de medios no comprimidos. Debe notarse que el flujo de bits a decodificar puede ser recibido desde un dispositivo remoto localizado dentro de virtualmente cualquier tipo de red. Adicionalmente, el flujo de bits puede ser recibido desde hardware o software local. Finalmente, un sintetizador de medios 170 puede reproducir los flujos de medios no comprimidos, por ejemplo con un altavoz o una pantalla. El receptor 150, el decodificador 160, y el sintetizador de medios 170 pueden residir en el mismo dispositivo físico o pueden incluirse en dispositivos separados.

La escalabilidad en términos de tasa de bits, complejidad de decodificación, y tamaño de imagen es una propiedad deseable para ambientes heterogéneos y propensos a errores. Esta propiedad es deseable con el fin de contrarrestar limitaciones tales como restricciones en la tasa de bits, la resolución de visualización, el caudal de la red, y el poder de cómputo en un dispositivo receptor.

Deberá entenderse que, a pesar de que el texto y los ejemplos contenidos en la presente pueden describir específicamente un proceso de codificación, una persona con experiencia en la técnica entendería fácilmente que los mismos conceptos y principios también se aplican al proceso de decodificación correspondiente, y viceversa. Debe apreciarse que el flujo de bits a decodificar puede ser recibido desde un dispositivo remoto localizado dentro de virtualmente cualquier tipo de red. Adicionalmente, el flujo de bits puede ser recibido desde hardware o software local.

Los dispositivos de comunicación de la presente invención pueden comunicarse usando varias tecnologías de transmisión incluyendo, pero sin limitarse a, acceso múltiple por división de códigos (CDMA), sistema global para comunicaciones móviles (GSM), sistema universal de telecomunicaciones móviles (UMTS), acceso múltiple por división de tiempo (TDMA), acceso múltiple por división de frecuencia (FDMA), protocolo de control de Transmisión/Protocolo de Internet (TCP/IP), sistema de mensajes cortos (SMS), servicio de mensajes multimedia (MMS), correo electrónico, servicio de mensajería instantánea (IMS), Bluetooth, IEEE 802.11, etc. Un dispositivo de comunicación puede comunicarse usando varios medios incluyendo, pero sin limitarse a radio, infrarrojos, láser, conexión por cable, y similares.

Las figuras 5 y 6 muestran un dispositivo móvil 12 representativo, en el cual puede implementarse la presente invención. Sin embargo, debe entenderse que la presente invención no pretende limitarse a un tipo particular de dispositivo móvil 12 o a otro dispositivo electrónico. Algunas de las características ilustradas en las figuras 5 y 6 podrían incorporarse en cualquiera o en todos los dispositivos que pueden ser utilizados en el sistema mostrado en la figura 4.

El dispositivo móvil 12 de las figuras 5 y 6 incluye un alojamiento 30, una pantalla 32 en forma de una pantalla de cristal líquido, un teclado 34, un micrófono 36, un auricular 38, una batería 40, un puerto de infrarrojos 42, una antena 44, una tarjeta inteligente 46 en forma de un UICC de conformidad con una realización de la invención, un lector de tarjetas 48, un circuito de interfaz de radio 52, un circuito de codificación 54, un controlador 56 y una memoria 58. Los circuitos y elementos individuales son todos de un tipo bien conocido en la técnica, por ejemplo, en el rango de dispositivos móviles de Nokia.

La presente invención proporciona un sistema y un procedimiento mejorados para implementar una administración eficiente de la memoria intermedia de imágenes decodificadas en la codificación de video de vistas múltiples. Para

5 tratar la cuestión en torno al hecho de que la sintaxis de JMVM no incluye la señalización de una imagen de la cual pueda iniciarse la decodificación de cierta vista (a menos que todas las vistas del índice de tiempo contengan una imagen ancla), una nueva señal se señalizada indicando si puede accederse a una vista desde cierta imagen, es decir, si la decodificación de una vista puede iniciarse desde cierta imagen. En una realización de la invención, esta señal se señala en el encabezado de unidades NAL. Lo siguiente es un ejemplo de la sintaxis y la semántica de la señal de conformidad con una realización particular. Sin embargo, también es posible cambiar la semántica del elemento de sintaxis `anchor_pic_flag` de manera similar en lugar de agregar un nuevo elemento de sintaxis.

	C	Descriptor
<code>nal_unit_header_svc_mvc_extension () {</code>		
<code>  <b>svcmvc_flag</b></code>	Todos	u(1)
<code>  if(!svc_mvc_flag){</code>		
<code>    <b>priority_id</b></code>	Todos	u(6)
<code>    <b>discardable_flag</b></code>	Todos	u(1)
<code>    <b>temporal_level</b></code>	Todos	u(3)
<code>    <b>dependency_id</b></code>	Todos	u(3)
<code>    <b>quality_level</b></code>	Todos	u(2)
<code>    <b>layer_base_flag</b></code>	Todos	u(1)
<code>    <b>use_base_prediction_flag</b></code>	Todos	u(1)
<code>    <b>fragmented_flag</b></code>	Todos	u(1)
<code>    <b>last_fragment_flag</b></code>	Todos	u(1)
<code>    <b>fragment_order</b></code>	Todos	u(2)
<code>    <b>reserved_zero_twobits</b></code>	Todos	u(2)
<code>  }else{</code>		
<code>    <b>view_refresh_flag</b></code>	Todos	u(1)
<code>    <b>view_subset_id</b></code>	Todos	u(2)
<code>    <b>view_level</b></code>	Todos	u(3)
<code>    <b>anchor_pic_flag</b></code>	Todos	u(1)
<code>    <b>view_id</b></code>	Todos	u(10)
<code>    <b>reserved_zero_five_bits</b></code>	Todos	u(6)
<code>  }</code>		
<code>  nalUnitHeaderBytes += 3</code>		
<code>}</code>		

10 Para cierta imagen en una vista, todas las imágenes en el mismo sitio temporal de otras vistas que utilizan la predicción inter-vistas se denominan "imágenes de vistas de dependencia directa", y todas las imágenes en el mismo sitio temporal de otras vistas que se requieren decodificar la imagen actual se denominan "imágenes de dependencia".

15 La semántica de `view_refresh_flag` puede especificarse en cuatro formas en una realización. Una primera forma de especificar la semántica de `view_refresh_flag` implica que `view_refresh_flag` indique que la imagen actual y todas las imágenes posteriores en el orden de salida en la misma vista pueden decodificarse correctamente cuando todas las imágenes de vista de dependencia directa de las imágenes actuales y posteriores en la misma vista son también (posiblemente en forma parcial) decodificadas sin decodificar ninguna imagen precedente en la misma vista o en otras vistas. Esto implica que (1) ninguna de las vistas de dependencia recae en ninguna imagen precedente en el orden de decodificación en ninguna vista, o (2) si cualquiera de las imágenes de vistas de dependencia recaen en cualquier imagen precedente en el orden de decodificación en cualquier vista, entonces solo las áreas intra-codificadas de manera restringida de las imágenes de vistas de dependencia directa de las imágenes actuales y posteriores en la misma vista se usan para predicción inter-vistas. Un área intra-codificada de manera restringida no utiliza datos de las áreas vecinas intercodificadas para la intra-predicción.

25 Una segunda forma de especificar la semántica de `view_refresh_flag` implica que `view_refresh_flag` indique que la imagen actual y todas las imágenes posteriores en el orden de decodificación en la misma vista pueden decodificarse correctamente cuando todas las imágenes de vistas de dependencia directa de la imagen actual e imágenes posteriores en la misma vista también son decodificadas completamente o, en una realización, parcialmente sin decodificar ninguna imagen precedente.

30 Una tercera forma de especificar la semántica de `view_refresh_flag` implica que `view_refresh_flag` indique que la imagen actual y todas las imágenes posteriores en el orden de salida en la misma vista pueden decodificarse correctamente cuando todas las imágenes de vistas de dependencia de las imágenes actuales y posteriores en la misma vista también son decodificadas completamente o, en una realización, parcialmente. Esta definición es análoga a una intra imagen que inicia un GOP abierto en la codificación de vista simple. En términos de texto de la memoria, esta opción puede escribirse como sigue: Una `view_refresh_flag` igual a 1 indica que la imagen actual y cualquier imagen posterior en el orden de decodificación en la misma vista que la imagen actual y siguiendo la imagen actual en el orden de salida no se refiere a una imagen que precede a la imagen actual en el orden de decodificación en el proceso de interpredicción. Una `view_refresh_flag` igual a 0 indica que la imagen actual o una

imagen posterior en el orden de decodificación en la misma vista que la imagen actual y siguiendo la imagen actual en el orden de salida puede referirse a una imagen que precede a la imagen actual en el orden de decodificación en el proceso de interpredicción.

5 Una cuarta forma de especificar la semántica de `view_refresh_flag` implica que `view_refresh_flag` indique que la imagen actual y todas las imágenes posteriores en el orden de decodificación en la misma vista pueden decodificarse correctamente cuando todas las imágenes de vistas de dependencia de las imágenes actuales y posteriores en la misma vista también son decodificadas completamente o, en una realización, parcialmente. Esta definición es análoga a una intra imagen que inicia un GOP cerrado en la codificación de vista simple.

10 La `view_refresh_flag` puede usarse en un sistema tal como el ilustrado en la figura 4. En esta situación, el receptor 150 ha recibido, o el decodificador 160 ha decodificado, sólo cierto subconjunto M de todas las N vistas disponibles, excluyendo el subconjunto la vista A. Debido a una acción de usuario, por ejemplo, al receptor 150 o al decodificador 160 le gustaría recibir o decodificar, respectivamente, la vista A de ahora en adelante. El decodificador puede comenzar la decodificación de la vista A de la primera imagen, con `view_refresh_flag` igual a 1 en la vista A. Si la vista A no fue recibida, entonces el receptor 150 puede indicar a la pasarela 140 o al emisor 130 que incluya imágenes codificadas de la vista A en el flujo de bits transmitido. La pasarela 140 o el emisor 130 pueden esperar hasta que la siguiente imagen tenga `view_refresh_flag` igual a 1 en la vista A antes de enviar alguna imagen de la vista A con el fin de evitar el envío de imágenes innecesarias de la vista A que el decodificador 160 no podría decodificar con éxito.

Para tratar la segunda cuestión discutida anteriormente, se señala una nueva señal para indicar si una vista se utiliza para referencia de predicción inter-vistas, y el elemento de sintaxis `nal_ref_idc` sólo indica si una imagen se usa para referencia de predicción temporal. En una realización particular, esta señal se señala en el encabezado de unidades NAL. El siguiente es un ejemplo de la sintaxis y la semántica de la señal.

	C	Descriptor
<code>nal_unit_header_svc_mvc_extension () {</code>		
<code>  <b>svcmvc_flag</b></code>	Todos	u(1)
<code>  if(!<code>svc_mvc_flag</code>){</code>		
<code>    <b>priority_id</b></code>	Todos	u(6)
<code>    <b>discardable_flag</b></code>	Todos	u(1)
<code>    <b>temporal_level</b></code>	Todos	u(3)
<code>    <b>dependency_id</b></code>	Todos	u(3)
<code>    <b>quality_level</b></code>	Todos	u(2)
<code>    <b>layer_base_flag</b></code>	Todos	u(1)
<code>    <b>use_base_prediction_flag</b></code>	Todos	u(1)
<code>    <b>fragmented_flag</b></code>	Todos	u(1)
<code>    <b>last_fragment_flag</b></code>	Todos	u(1)
<code>    <b>fragment_order</b></code>	Todos	u(2)
<code>    <b>reserved_zero_twobits</b></code>	Todos	u(2)
<code>  }else{</code>		
<code>    <b>view refresh flag</b></code>	Todos	u(1)
<code>    <b>view_subset_id</b></code>	Todos	u(2)
<code>    <b>view_level</b></code>	Todos	u(3)
<code>    <b>anchor_pic_flag</b></code>	Todos	u(1)
<code>    <b>view_id</b></code>	Todos	u(10)
<code>    <b>reserved_zero_five_bits</b></code>	Todos	u(6)
<code>  }</code>		
<code>  nalUnitHeaderBytes += 3</code>		
<code>}</code>		

30 Una `inter_view_reference_flag` igual a 0 indica que la imagen actual no se usa como una imagen de referencia inter-vistas. Una `inter_view_refresh_flag` igual a 1 indica que la imagen actual se usa como imagen de referencia inter-vistas. Se infiere que el valor de `inter_view_refresh_flag` es igual a 1 cuando `profile_idc` indica un perfil de MVC y `view_id` es 0. Cuando se decodifica una imagen, todas las imágenes que tienen una `inter_view_refresh_flag` igual a 1 y con el mismo eje temporal como la imagen actual se denominan imágenes inter-vistas de la imagen actual.

35 La `inter_view_refresh_flag` puede usarse en una pasarela 140, también referida como elemento de red de conocimiento de medios (MANE). Cuando una imagen no es utilizada como referencia inter-vistas y referencia intra-vistas (`inter_view_refresh_flag` es igual a 0 y `nal_ref_idc` es igual a 0), un MANE puede elegir no enviarlo sin consecuencias en la decodificación del flujo de bits remanente. Cuando una imagen no se utiliza como referencia inter-vistas pero se usa como una referencia intra-vistas, un MANE debe depositar la imagen sólo si también deposita la transmisión de las vistas dependientes. Cuando una imagen no se utiliza como una referencia inter-vistas, pero se usa como una referencia intra-vistas, un MANE debe depositar la imagen sólo si no se requiere o desea decodificar la vista en donde reside la imagen.

Con respecto a la cuestión del proceso de marcación de imágenes de referencia especificado en JMVM 1.0 que no es capaz de manejar eficientemente la administración de imágenes decodificadas que deben almacenarse temporalmente para la predicción inter-vistas, se reutiliza la señal `inter_view_refresh_flag`. Las imágenes con una `inter_view_refresh_flag` igual a 1 pueden marcarse usando cualquiera de tres procedimientos.

Un primer procedimiento para marcar imágenes con una `inter_view_refresh_flag` igual a 1 incluye almacenar imágenes de referencia inter-vistas temporalmente como imágenes de largo plazo. En el proceso de codificación, cada imagen utilizada para la predicción inter-vistas está indicada en el flujo de bits para ser marcada como "usada para referencia de largo plazo". Una forma de indicar la marcación como "usada para referencia de largo plazo" es la `inter_view_refresh_flag`. El decodificador responde a la indicación marcando la imagen como "usada para referencia de largo plazo" y "referencia temporal de largo plazo de vistas múltiples". Cualquier operación de control de administración de memoria dirigida a una imagen marcada como "usada para referencia de largo plazo" y "referencia temporal de largo plazo de vistas múltiples" es almacenada temporalmente. Cuando todas las imágenes en el eje temporal son codificadas o decodificadas, todas las imágenes marcadas como "usadas para referencia de largo plazo" y "referencia temporal de largo plazo de vistas múltiples" ya no están marcadas como "usadas para referencia de largo plazo" y "referencia temporal de largo plazo de vistas múltiples", y la marcación de imágenes de referencia se vuelve a realizar en su orden de decodificación usando ya sea la operación de ventana deslizante u operaciones de control de administración de memoria intermedia (cualquiera que se aplique a una imagen particular). Por ejemplo, si se usa una imagen para interpredicción (es decir, el valor de `nal_ref_idc` es mayor que 0), se vuelve a marcar como "usadas para referencia de corto plazo". Si la imagen no se utiliza para interpredicción (es decir, `nal_ref_idc` es igual a 0), se marca como "no usada para referencia". Usualmente, solo hay dos casos para la imagen en cierto eje temporal: todas las imágenes son imágenes de referencia para interpredicción, o ninguna imagen es una imagen de referencia para interpredicción. Esta última operación puede realizarse después de que se ha decodificado la última unidad VCL NAL en el eje temporal, o antes de que la siguiente unidad de acceso o la siguiente imagen en el eje temporal posterior sea decodificada. En el proceso de decodificación, la operación en esta etapa puede activarse implícitamente por el cambio en el eje temporal, o puede señalizarse explícitamente por ejemplo, como un comando MMCO. Con este procedimiento, las imágenes de referencias inter-vistas tienen la misma influencia que las imágenes de referencia de largo plazo para la predicción ponderada y en el modo directo temporal.

Un segundo procedimiento para marcar imágenes con una `inter_view_refresh_flag` igual a 1 incluye marcar las imágenes de referencia inter-vistas como "usadas para referencia inter-vistas". Con este procedimiento la marcación de la imagen de referencia para la interpredicción (marcación como "usada para referencia de corto plazo" y "usada para referencia de largo plazo") permanece sin cambio en comparación con el estándar AVC. Para procesos relacionados con el modo directo temporal y la predicción ponderada, las imágenes marcadas como "usadas para referencia inter-vistas", es decir, aquellas imágenes de referencia inter-vistas que comparten el mismo eje temporal que la imagen actual, son tratadas idénticamente a las imágenes de referencia de largo plazo. Cuando todas las imágenes en el eje temporal son codificadas o decodificadas, todas las imágenes marcadas como "usadas para referencia inter-vistas" ya no están marcadas como "usadas para referencia inter-vistas".

Se aprecia que la retirada de la marcación "usadas para referencia inter-vistas" después de que son procesadas todas las imágenes en el eje temporal es solo una realización de la invención. La marcación como "usada para referencia inter-vistas" también podría retirarse en otros instantes del proceso de decodificación. Por ejemplo, la marcación como "usada para referencia inter-vistas" de una imagen particular puede retirarse tan pronto la imagen actual o cualquier imagen posterior deje de depender directa o indirectamente de la imagen de acuerdo con la señalización de dependencia de la vista incluida en la extensión MVC de SPS.

La operación de ya no marcar las imágenes apropiadas como "usadas para referencia inter-vistas" puede realizarse después de que la última unidad VCL NAL en el eje temporal se decodifica o antes de que se decodifique la siguiente unidad de acceso o la siguiente imagen en el eje temporal posterior. En el proceso de decodificación, esto puede activarse implícitamente por el cambio en el eje temporal o puede señalizarse explícitamente, por ejemplo, como un comando MMCO.

Con este procedimiento particular, las imágenes de referencia inter-vistas tienen la misma influencia que las imágenes de referencia de largo plazo para la predicción ponderada y en el modo directo temporal. En otras palabras, este procedimiento tiene el mismo efecto que el primer procedimiento discutido líneas arriba para la predicción ponderada y en el modo directo temporal.

En el presente procedimiento, puede aplicarse un mecanismo de ventana deslizante mejorado para retirar la marcación de "usadas para referencia inter-vistas" de imágenes utilizadas solo para predicción inter-vistas, es decir imágenes que tienen `nal_ref_idc` igual a 0 y marcadas como "usadas para referencia inter-vistas". Este mecanismo de ventana deslizante mejorado utiliza una variable, por ejemplo, llamada `num_inter_view_ref_frames`, señalado preferentemente en la extensión SPS para MVC, de tal forma que cuando el número de imágenes marcadas como "usadas para referencia inter-vistas" y que tienen `nal_ref_idc` igual a 0 es igual a `num_inter_view_ref_frames`, entonces la que se decodificó primero se convierte en no marcada como "usada para referencia inter-vistas".

Consecuentemente, si la imagen tampoco necesita enviarse (enviarse ya o no sacarse intencionalmente), el decodificador puede invocar un proceso para retirar la imagen del DPB de tal forma que puede almacenarse en el DPB una imagen recientemente decodificada.

5 Un tercer procedimiento para marcar imágenes con una `inter_view_reference_flag` igual a 1 incluye marcar imágenes después de decodificar todas las imágenes del mismo eje temporal/índice de tiempo. En lugar de marcar una imagen inmediatamente después de su decodificación, este procedimiento se basa en la idea de que las imágenes son marcadas después de decodificar todas las imágenes del mismo eje temporal (es decir, el mismo índice temporal). La ventana deslizante o la marcación de imágenes de referencia adaptativa como se indica en cada una de las imágenes codificadas se realiza en el orden en el que se decodificaron las imágenes. Para procesos relacionados con el modo directo temporal y la predicción ponderada, las imágenes marcadas del mismo eje temporal como la imagen actual son tratadas de manera idéntica a las imágenes de referencia de largo plazo. Las imágenes de referencia inter-vistas del mismo eje temporal que la imagen actual están incluidas en la construcción de las listas de imágenes de referencia inicial y pueden reordenarse con base en su `view_id` o se les asignan primero índices de referencia de largo plazo, y pueden entonces volver a correlacionarse con base en el índice de referencia de largo plazo.

Como se discutió anteriormente, dada la manera de recalcularse el `PicNum`, si el modo de operación de ventana deslizante está en uso y el número de imágenes de corto plazo y de largo plazo es igual al máximo, la imagen de referencia de corto plazo que tiene el `FrameNumWrap` más pequeño se marca como "no usado para referencia". Sin embargo, debido al hecho de que esta imagen no es necesariamente la imagen codificada más anterior porque el orden del `FrameNum` en el JMVM actual no sigue el orden de decodificación, la marcación de la imagen de referencia de ventana deslizante no opera en forma óptima en el JMVM actual. Para resolver este problema, y en comparación con el estándar JMVM, los `FrameNum` y `FrameNumWrap` variables no son redefinidos/escalados, es decir, su definición se mantiene sin cambios comparado con el estándar AVC. Está diseñado que las imágenes de corto plazo puedan ser administradas automáticamente por el mecanismo primero en entrar, primero en salir de la ventana deslizante. Solo se requiere una modificación ligera del mecanismo de ventana deslizante comparado con el JMVM 1.0. Las modificaciones son como sigue, con el nuevo texto representado en *itálica*:

30 **G.8.2.5.3 Proceso de marcación de imágenes de referencia decodificadas de ventana deslizante**  
 Este proceso es invocado cuando `adaptive_ref_pic_marking_mode_flag` es igual a 0. *Sólo las imágenes de referencia que tienen la misma `view_id` que la línea actual se consideran que están en el proceso, incluyendo el cálculo de `numShortTerm` y `numLongTerm`, y el valor aplicado de `num_ref_frames`.*

35 En el procedimiento anterior, el número total de cuadros de referencia para todo el flujo de bits de MVC, que indica el tamaño de la memoria intermedia para el almacenamiento de imágenes utilizadas para referencia intra-vistas o inter-vistas de todo un flujo de bits de MVC, debe ser igual a la suma de los valores de `num_ref_frames` aplicados para todas las vistas contenidas en el flujo de bits de MVC más el número máximo de cuadros de referencia inter-vistas para decodificar el flujo de bits de MVC. Alternativamente, la ventana deslizante puede realizarse globalmente para todas las imágenes en todas las vistas.

Para la codificación de primero el tiempo, el proceso de ventana deslizante se define como sigue, con nuevo texto para JMVM 1.0 representado en *itálica*:

45 **G.8.2.5.3 Proceso de marcación de imágenes de referencia decodificadas de ventana deslizante**  
 ...  
 ...  
 - Cuando `numShortTerm + numLongTerm` es igual a  $\text{Max}(\text{num\_ref\_frames}, 1)$ , deberá cumplirse la condición de que `numShortTerm` es mayor que 0, y el cuadro de referencia de corto plazo, el par de campos de referencias complementarias o campo de referencias no pares que *se selecciona mediante la siguiente regla se marca como "no usado para referencia"*. Cuando es un cuadro o un par de campos complementarios, ambos campos son marcados también como "no usados para referencia".  
 \**La regla de selección es: de todas aquellas imágenes con el valor más pequeño de `FrameNumWrap`, se selecciona la primera en el orden de decodificación. El orden de decodificación de esas imágenes puede indicarse por el valor de `view_id`, o la información de dependencia de vistas señalizadas en el SPS de la extensión de MVC.*

60 Para la codificación de primero el tiempo, el proceso de ventana deslizante se define como sigue, con nuevo texto para JMVM 1.0 representado en *itálica*:

65 **G.8.2.5.3 Proceso de marcación de imágenes de referencia decodificadas de ventana deslizante**  
 ...  
 ...  
 - Cuando `numShortTerm + numLongTerm` es igual a  $\text{Max}(\text{num\_ref\_frames}, 1)$ , deberá cumplirse la

condición de que numShortTerm es mayor que 0, y el cuadro de referencia de corto plazo, el par de campos de referencias complementarios o campo de referencias no pares que *se selecciona mediante la siguiente regla* se marca como "no usado para referencia". Cuando es un cuadro o un par de campos complementarios, ambos campos son marcados también como "no usados para referencia".

5 *\*La regla de selección es: de todas aquellas imágenes de la vista decodificada más anterior, se selecciona aquella con el valor de FrameNumWrap más pequeño. El orden de decodificación de las vistas puede indicarse por el valor de view\_id, a la información de dependencia de vistas señalizada en el SPS de la extensión de MVC.*

10 Como se discutió anteriormente, debido al hecho de que PicNum se deriva del FrameNumWrap redefinido y escalado, la diferencia entre los valores de PicNum de las dos imágenes codificadas se escalaría en promedio. Por ejemplo, es de utilidad suponer que existen dos imágenes en la misma vista y que tienen frame\_num igual a 3 y 5, respectivamente. Cuando sólo hay una vista, es decir, el flujo de bits es un flujo AVC, entonces la diferencia de los dos valores de PicNum sería 2. Cuando se codifica la imagen que tiene frame\_num igual a 5, si se necesita un comando MMCO para marcar la imagen que tiene PicNum igual a 3 como "no usado para referencia", entonces la diferencia de los dos valores menos 1 es igual a 1, lo cual se señalará en el MMCO. Este valor necesita 3 bits. Sin embargo, si hay 256 vistas, entonces la diferencia de los dos valores PicNum menos 1 sería 511. En este caso, se requieren 19 bits para señalar el valor. Consecuentemente, los comandos MMCO son codificados con mucha menos eficiencia. Normalmente, el mayor número de bits es igual a  $2 \cdot \log_2$  (número de vistas) para un comando MMCO del JMVM actual comparado con la codificación de vista simple de H.264/AVC.

15 Para resolver este problema y en contraste con el estándar JMVM, las variables FrameNum y FrameNumWrap no son redefinidas/escaladas, lo cual es lo mismo que en el estándar AVC. En la mayoría de los casos, no se requiere desde el punto de vista del tamaño de DPB que una imagen contenga un comando MMCO para retirar una imagen que no pertenece a la misma vista ni pertenece al mismo eje temporal que la imagen actual. Incluso algunas de las imágenes dejan de ser necesarias para referencia y, por lo tanto, pueden marcarse como "no usadas para referencia". En este caso, la marcación puede realizarse usando el proceso de ventana deslizante o posponerse hasta la siguiente imagen codificada con la misma view\_id. Por lo tanto, los comandos de MMCO están restringidos sólo para marcar imágenes como "no usadas como referencia" para imágenes que pertenecen a la misma vista o al mismo eje temporal, aunque el DBP puede contener imágenes de diferentes vistas o diferentes ejes temporales.

20 La modificación de JMVM 1.0 para marcación de imágenes de referencia intra-vistas es como sigue, con los cambios mostrados en *itálica*:

35 G.8.2.5.4.1 Proceso de marcación de una imagen de referencia de corto plazo como "no usada como referencia"  
Este proceso es invocado cuando *adaptive\_pic\_marking\_mode\_flag* es igual a 1. *Sólo las imágenes de referencia que tienen la misma view\_id que la línea actual se consideran en este proceso.*

40 La sintaxis y semántica para la marcación de imágenes de referencia puede ser como sigue:

slice_header () {	C	Descriptor
...		
if(nal_ref_ido !=0)		
dec_ref_pic_marking ()	2	
if(inter_view_reference_flag)		
dec_view_ref_pic_marking_mvc ()	2	
}		

dec_view_ref_pic_marking_mvc () {	C	Descriptor
<b>adaptive_view_ref_pic_marking_mode_flag</b>	2	u(1)
if(adaptive_view_ref_pic_marking_mode_flag)		
do {		
<b>view_memory_management_control_operation</b>	2	ue(v)
if(view_memory_management_control_operation == 1    view_memory_management_control_operation == 2)		
<b>abs_difference_of_view_id_minus1</b>	2	ue(v)
} while(view_memory_management_control_operation != 0)		
}		
}		

Operación de control de gestión de administración de memoria

45 Los valores de la operación de control de administración de memoria (view\_memory\_management\_control\_operation) son los siguientes

view_memorymanagement_control_operation	Operación de control de administración de memoria
0	Terminar el ciclo view_memory_management_controloperation
1	Retirar la marcación de "usado para referencia entre vistas" o marcar una imagen como "no usada para referencia", abs_difference_of_view_id_minus1 está presente y corresponde a una diferencia para abstraer de view_id actual
2	Retirar la marcación de "usado para referencia entre vistas" o marcar una imagen como "no usada para referencia", abs_difference_of_view_id_minus1 está presente y corresponde a una diferencia para agregar a view_id actual

La adaptive\_view\_ref\_pic\_marking\_mode\_flag especifica si está en uso el mecanismo de ventana deslizante (cuando es igual a 0) o el proceso de marcación de imágenes de referencia adaptativo (cuando es igual a 1).

5 El proceso de decodificación modificado para la marcación de imágenes de referencia inter-vistas es como sigue:

#### 8.2.5.5.2 Marcación de imágenes inter-vistas

Este proceso es invocado cuando view\_memory\_management\_control\_operation es igual a 1.

10

Especifíquese viewIDX de la siguiente manera.

```

if(view memory management control_operation==1)
  viewIDX = CurrViewid -(difference_of_view_id_minus 1 + 1)
15 else if(view_memory_management_control_operation == 2)
  viewIDX = CurrViewid +(difference_of_view_id_minus + 1)

```

20

Para permitir la escalabilidad, es decir, la posibilidad de elegir qué vistas son transmitidas, enviadas, o decodificadas, las operaciones de control de administración de memoria pueden restringirse de la siguiente manera. Si currTemporalLevel es igual a temporal\_level de la imagen actual y dependentViews es un conjunto de vistas que dependen de la vista actual, un comando MMCO puede solamente dirigirse a una imagen que tiene un temporal\_level igual a o mayor que currTemporalLevel y está dentro de dependentViews. Para permitir esto, los comandos MMCO son anexados con una indicación de view\_id o se especifican nuevos comandos MMCO con una indicación de view\_id.

25

Con el fin de resolver estos problemas relacionados con el proceso de construcción de listas de imágenes de referencia descrito anteriormente, los FrameNum y FrameNumWrap variables no son redefinidos/escalados. Esta es la misma acción que se produce en el estándar AVC y contrasta con el estándar JMVM, donde las variables son redefinidas/reescaladas. La modificación de JMVM 1.0 es como se muestra a continuación, con cambios mostrados en *italica*:

30

En 8.2.4.3.1, el proceso de reordenamiento de las listas de imágenes de referencia para imágenes de referencia de corto plazo, 8-38 deberá cambiarse como:

35

```

for (cldx = num_ref_idx_IX_active_minus1 + 1; cldx > refldxLX; cldx-- )
  RefPicListX[cldx] = RefPicListX[cldx - 1]
  RefPicListX[refldxLX++] = imagen de referencia de corto plazo con PicNum igual a picNumLX y view_id igual a CurrViewID
  nldx = ref_idxLX
40 for(cldx = refldxLX; cldx <= num_ref_idx_1X_active_minus1 + 1; cldx++) (8-38)
  //if (PicNumF (RefPicListX rcldx) ) !=picNumLX)
  if (PicNumF (RefPicListX [cldbx]) != picNumLX || ViewID (RefPicListX [cldx] != CurrViewID)
  RefPicListX[nldx++] = RefPicListX[cldx]

```

45

Donde CurrViewID es la view\_id de la imagen de decodificación actual.

50

Respecto a los problemas asociados con el proceso de inicialización de listas de imágenes de referencia discutido anteriormente, estos problemas pueden resolverse apreciando que sólo cuadros, campos, o pares de campos que pertenecen a la misma vista como la línea actual pueden considerarse en el proceso de inicialización. En términos de JMVM 1.0, este lenguaje puede agregarse al principio de cada una de las subclases 8.2.4.2.1 "Proceso de inicialización para la lista de imágenes de referencia para líneas P y SP en cuadros" a través de 8.2.4.2.5 "Proceso de inicialización para listas de imágenes de referencia en los campos".

Con respecto a otras cuestiones que se relacionan con el proceso de construcción de listas de imágenes de referencia, pueden usarse varios procedimientos para reordenar eficientemente tanto imágenes inter-vistas e imágenes utilizadas para intra-predicción. Un primer procedimiento de tales procedimientos involucra colocar en la lista imágenes de referencia inter-vistas frente a imágenes de referencia intra-vistas, así como especificar procesos RPLR separados para imágenes inter-vistas e imágenes para predicción intra-vistas. Las imágenes utilizadas para predicción intra-vistas también se denominan imágenes intra-vistas. En este procedimiento, se realiza el proceso de inicialización de listas de imágenes de referencia para imágenes intra-vistas como se especifica anteriormente, seguido por el proceso de reordenamiento de RPLR y el proceso de truncamiento de lista para las imágenes intra-vistas. Enseguida, las imágenes inter-vistas son anexadas a la lista después de las imágenes intra-vistas. Finalmente, cada imagen inter-vistas puede seleccionarse adicionalmente y colocarse en una entrada específica de la lista de imágenes de referencia usando la siguiente sintaxis, semántica y proceso de decodificación, modificado de JMVM 1.0. El procedimiento es aplicable tanto a refPicList0 como a refPicList1, si está presente.

	C	Descriptor
ref_pic_list_reordering () {		
if(slice_type != I && slice_type != SI) {		
...		
}		
if(svc_mvc_flag)		
{		
<b>view_ref_pic_list_reordering_flag_10</b>	2	u(1)
if(view_ref_pic_list_reordering_flag_10)		
do {		
<b>view_reordering_idc</b>	2	ue(v)
if (view_reordering_idc == 0    view_reordering_idc == 1)		
<b>abs_diff_view_idx_minus1</b>	2	ue(v)
<b>ref_idx</b>	2	ue(v)
} while (view_reordering_idc != 2)		
}		

Con respecto a la sintaxis, una `view_ref_pic_list_reordering_flag_IX` (X es 0 ó 1) igual a 1 especifica que el elemento de sintaxis `view_reordering_idc` está presente para `refPicListX`. Una `view_ref_pic_list_reordering_flag_IX` igual a 0 especifica que el elemento de sintaxis `view_reordering_idc` no está presente para `refPicListX`. La `ref_idx` indica la entrada que la imagen inter-vistas va a colocar en la lista de imágenes de referencia.

El `abs_diff_view_idx_minus1` más 1 especifica la diferencia absoluta entre el índice de vistas de la imagen que se colocará a la entrada de la lista de imágenes de referencia indicada por `ref_idx` y el valor de predicción del índice de vistas. `abs_diff_view_idx_minus1` se encuentra en el intervalo de 0 a `num_multiview_refs_for_listX[view id]-1`. El `num_multiview_refs_for_listX[]` se refiere a `anchor_reference_view_for_list_X[curr_view_id] []` para una imagen ancla `non_anchor_reference_view_for_list_X[curr_view_id] []` para una imagen no ancla, en donde `curr_view_id` es igual a `view_id` de la vista que contiene la línea actual. Un índice de vistas de una imagen inter-vistas indica el orden de `view_id` de la imagen inter-vistas que aparece en la extensión MVC SPS. Para una imagen con un índice de vistas igual a `view_index`, `view_id` es igual a `num_multiview_refs_for_listX[view index]`.

El `abs_diff_view_idx_minus1` más 1 especifica la diferencia absoluta entre el índice de vistas de la imagen que está siendo movida al índice actual en la lista y el valor de predicción de índices de listas. `abs_diff_view_idx_minus1` se encuentra en el intervalo de 0 a `num_multiview_refs_for_listX[view id] -1`. El `num_multiview_refs_for_listX[]` se refiere a `anchor_reference_view_for_list_X[curr_view_id] []` para una imagen ancla y `non_anchor_reference_view_for_list_X[curr_view_id] []` para una imagen no ancla, donde `curr_view_id` es igual a `view_id` de la vista que contiene la línea actual. Un índice de vistas de una imagen inter-vistas indica el orden de `view_id` de la imagen inter-vistas que aparece en la extensión MVC SPS. Para una imagen con un índice de vistas igual a `view_index`, `view_id` es igual a `num_multiview_refs_for_listX [view_index]`.

El proceso de decodificación es el siguiente:

La definición de `NumRefIdxLXActive` se realiza después del truncamiento para imágenes intra-vistas:

$$\text{NumRefIdxLXActive} = \text{num\_ref\_idx\_IX\_active\_minus1} + 1 + \text{num\_multiview\_refs\_for\_listX}[\text{view id}]$$

G.8.2.4.3.3 Proceso de reordenamiento de listas de imágenes de referencia para imágenes inter-vistas. Las entradas a este proceso son la lista de imágenes de referencia `RefPicListX` (siendo X 0 ó 1). Las salidas de este proceso son una lista de imágenes de referencia posiblemente modificada `RefPicListX` (siendo X 0 ó 1).

La variable `picViewIdxLX` se deriva de la siguiente manera.

Si `view_reordering_idc` es igual a 0  
`picViewIdxLX = picViewIdxLXPred - (abs_diff_view_idx_minus1+1)`  
 De lo contrario (`view_reordering_idc` es igual a 1),  
`picViewIdxLX = picViewIdxLXPred + (abs_diff_view_idx_minus1+1)`  
 5 `picViewIdxLXPred` es el valor de predicción para la variable `picViewIdxLX`. Cuando el proceso especificado en esta subcláusula es invocado por primera vez para una línea (es decir, para la primera aparición de `view_reordering_idc` igual a 0 ó 1 en la sintaxis de `ref_pic_list_reordering()`), `picViewIdxL0Pred` y `picViewIdxL1Pred` se fijan inicialmente igual a 0. Después de cada asignación de `picViewIdxLX`, el valor de `picViewIdxLX` es asignado a `picViewIdxLXPred`.  
 El siguiente procedimiento se lleva a cabo para colocar la imagen inter-vistas con el índice de vistas igual a `picViewIdxLX` en la posición del índice, `ref_idx` cambia la posición de cualquier otra imagen restante para después  
 10 en la lista, de la siguiente manera.  
 For (`cldx = NumRefIdxLXActive`; `c_dx > ref_idx`; `cldx--`)  
     `RefPicListX[cldx] = RefPicListX[cldx - 1]`  
`RefPicListX[ref_idx]` = imagen de referencia inter-vistas con id de vista igual a  
 15 `Reference_view_for_list_X[picViewIdxLX]`  
     `nIdx = ref_idx + 1;`  
     for (`cldx = ref_idxLX`; `cldx <= NumRefIdxLXActive`; `cldx++`)  
         if (`ViewID(RefPicListX[cldx]) != TargetViewID1Time(RefPicListX[cldx]) != TargetTime`)  
             `RefPicListX[nIdx++] = RefPicListX[cldx]`  
 20 `preView_id = PicViewIDLX`  
     `TargetViewID` y `TargetTime` indican la `view_id` o el valor de eje temporal de la imagen de referencia objetivo que se reordenará, y `Time(pic)` regresa el valor del eje temporal de la imagen pic.

De conformidad con un segundo procedimiento para reordenar eficientemente imágenes inter-vistas e imágenes  
 25 utilizadas para intra-predicción, se realiza el proceso de inicialización de listas de imágenes de referencia para imágenes intra-vistas como se especifica anteriormente, y las imágenes inter-vistas son anexadas al final de la lista en el orden en el que aparecen en la extensión MVC SPS. Enseguida, se aplica un proceso de reordenamiento de RPLR tanto para imágenes intra-vistas como inter-vistas, seguido por un proceso de truncamiento de lista. La sintaxis, la semántica y el proceso, como muestra de decodificación modificado con base en JMVM, son como sigue:

30 Sintaxis de reordenamiento de la lista de imágenes de referencia.

	C	Descriptor
<code>ref_pic_list_reordering () {</code>		
<code>if(slice_type != I &amp;&amp; slice_type != B) {</code>		
<b>ref_pic_list_reordering_flag_10</b>	2	u(1)
<code>if(ref_pic_list_reordering_flag_10)</code>		
do {		
<b>reordering_of_pic_nums_idc</b>	2	ue(v)
<code>if (reordering_of_pic_nums_idc == 0    reordering_of_pic_nums_idc == 1)</code>		
<b>abs_diff_pic_num_minus1</b>	2	ue(v)
<code>else if(reordering_of_pic_nums_idc == 2)</code>		
<b>long_term_pic_num</b>	2	ue(v)
<code>if(reordering_of_pic_nums_idc == 4    reordering_of_pic_nums_idc == 5)</code>		
<b>abs_diff_view_idx_minus1</b>	2	ue(v)
<code>} while (reordering_of_pic_nums_idc != 3)</code>		
}		
<code>if(slice_type == B    slice_type == EB) {</code>		
<b>ref_pic_list_reordering_flag_l1</b>	2	u(1)
<code>if(ref_pic_list_reordering_flag_l1)</code>		
do {		
<b>reordering_of_pic_nums_idc</b>	2	ue(v)
<code>if(reordering_of_pic_nums_idc == 0    reordering_of_pic_nums_idc == 1)</code>		
<b>abs_diff_pic_num_minus1</b>	2	ue(v)
<code>else if(reordering_of_pic_nums_idc == 2)</code>		
<b>long_term_pic_num</b>	2	ue(v)
<code>if(reordering_of_pic_nums_idc == 4    reordering_of_pic_nums_idc == 5)</code>		
<b>abs_diff_view_idx_minus1</b>	2	ue(v)
<code>} while(reordering_of_pic_nums_idc != 3)</code>		
}		
}		
}		
}		

G.7.4.3.1 Semántica de reordenamiento de listas de imágenes de referencia

Tabla

Operaciones reordering_of_pic_nums_idc para reordenar listas de imágenes de referencia	
reordering_of_pic_nums_idc	Reordenamiento especificado
0	abs_diff_pic_num_minus1 está presente y corresponde a una diferencia para sustraer de un valor de predicción de numero de imagen
1	abs_diff_pic_num_minus1 está presente y corresponde a una diferencia para adicionar a un valor de predicción de numero de imagen
2	long_term_pic_num está presente y especifica el número de imágenes de largo plazo para una imagen de referencia
3	Fin de ciclo para reordenar la lista de imágenes de referencia inicial
4	abs_diff_view_idx_minus1 está presente y corresponde a una diferencia para sustraer de un valor de predicción de índice de vistas
5	abs_diff_view_idx_minus1 está presente y corresponde a una diferencia para adicionar a un valor de predicción de índice de vistas

5 El reordering\_of\_pic\_nums\_idc, junto con abs\_diff\_pic\_num\_minus1 o long\_term\_pic\_num, especifica qué imágenes de referencia vuelven a representarse. El reordering\_of\_pic\_nums\_idc, junto con abs\_diff\_view\_idx\_minus1, especifica qué imágenes de referencia inter-vistas se vuelven a representar. Los valores de reordering\_of\_pic\_nums\_idc están especificados en la tabla anterior. El valor del primer reordering\_of\_pic\_nums\_idc que sigue inmediatamente después de ref\_pic\_list\_reordering\_flag\_10 o ref\_pic\_list\_reordering\_flag\_11 no es igual a 3.

15 El abs\_diff\_view\_idx\_minus1 más 1 especifica la diferencia absoluta entre el índice de vistas de la imagen para colocar en el índice actual en la lista de imágenes de referencia y el valor de predicción del índice de vistas. abs\_diff\_view\_idx\_minus1 se encuentra en el intervalo de 0 a num\_multiview\_refs\_for\_listX[view\_id]-1. num\_multiview\_refs\_for\_listX[] se refiere a anchor\_reference\_view\_for\_list\_X[curr\_view\_id] [] para una imagen ancla y non\_anchor\_reference\_view\_for\_list\_X[curr\_view\_id] [] para una imagen no ancla, en donde curr\_view\_id es igual a view\_id de la vista que contiene la línea actual. Un índice de vistas de una imagen inter-vistas indica el orden del view\_id de la imagen inter-vistas que aparece en la extensión MVC SPS. Para una imagen con un índice de vistas igual a view\_index, el view\_id es igual a num\_multiview\_refs\_for\_listX[view\_index].

El proceso de reordenamiento puede describirse como sigue:

25 G.8.2.4.3.3 Proceso de reordenamiento de listas de imágenes de referencia para imágenes inter-vistas.

La entrada a este proceso es un índice refIdxLX (siendo X 0 ó 1).  
 La salida de este proceso es un índice incrementado refIdxLX.  
 La variable picViewIdxLX se deriva como sigue.  
 Si reordering\_of\_pic\_nums\_idc es igual a 4  
 30 picViewIdxLX = picViewIdxLXPred - (abs\_diff\_view\_idx\_minus1 + 1)  
 De lo contrario (reordering\_of\_pic\_nums\_idc es igual a 5),  
 picViewIdxLX = picViewIdxLXPred + (abs\_diff\_view\_idx\_minus1 + 1)  
 picViewIdxLXPred es el valor de predicción para la variable picViewIdxLX. Cuando se invoca el proceso especificado en esta sub-cláusula la primera vez para una línea (es decir, para la primera aparición de  
 35 reordering\_of\_pic\_nums\_idc igual a 4 ó 5 en la sintaxis ref\_pic\_list\_reordering()), picViewIdxL0Pred y picViewIdxL1Pred se fijan inicialmente iguales a 0. Después de cada asignación de picViewIdxLX, el valor de picViewIdxLX se asigna a picViewIdxLXPred.

40 El siguiente procedimiento se lleva a cabo para colocar la imagen inter-vistas con un índice de vistas igual a picViewIdxLX dentro de la posición del índice refIdxLX, cambia la posición de cualquier otra imagen restante para después en la lista, e incrementa el valor de refIdxLX.

```

for (cldx = num_ref_idx_IX_active_minus1 + 1; cldx > refIdxLX; cldx--)
    RefPicListX[cldx] = RefPicListX[cldx-1]
45 RefPicListX[refIdxLX++] = imagen de referencia inter-vistas con id de vistas igual a
    Reference_view_for_listX[picViewIdxLX]
    
```

```

nldx = refldxLX
for (cldx = refldxLX; cldx <= num_ref_idx_IX_active_minus1 + 1; cldx++)
  if (ViewID(RefPicListX[cldx]) != TargetViewID || Time (RefPicListX[ cldx ]) != TargetTime)
    RefPicListX[nldx++] = RefPicListX[ cldx ]

```

5 Donde TargetViewID y TargetTime indican el view\_id o el valor del eje temporal de la imagen de referencia objetivo que va a reordenarse, y Time(pic) regresa el valor de eje temporal de la imagen pic.

10 De acuerdo con un tercer procedimiento para reordenar eficientemente tanto imágenes inter-vistas e imágenes utilizadas para intra-predicción, la lista de imágenes de referencia inicial contiene imágenes marcadas como "usadas como referencia de corto plazo" o "usadas como referencia de largo plazo" y que tienen la misma view\_id que la imagen actual. Además, la lista de imágenes de referencia inicial contiene las imágenes que pueden usarse para predicción inter-vistas. Las imágenes usadas para predicción inter-vistas se concluyen a partir de la extensión del conjunto de parámetros de secuencias para MVC y se pueden concluir también de inter\_view\_reference\_flag. A las imágenes para predicción inter-vistas se les asignan ciertos índices de referencia de largo plazo para el proceso de decodificación de esta imagen. Los índices de referencia de largo plazo asignados para imágenes de referencia inter-vistas pueden, por ejemplo, ser los primeros N índices de referencia, y los índices para imágenes de largo plazo intra-vistas pueden modificarse para ser igual a su valor anterior + N para el proceso de decodificación de esta imagen, donde N representa el número de imágenes de referencia inter-vistas. Como alternativa, los índices de referencia de largo plazo indicados pueden estar en el intervalo de MaxLongTermFrameldx + 1 a MaxLongTermFrameldx + N, inclusive. Como alternativa, la extensión del conjunto de parámetros de secuencias para MVC puede contener un elemento de sintaxis denominado aquí como start\_lt\_index\_for\_rplr, y los índices de largo plazo asignados asignan el intervalo start\_lt\_index\_for\_rplr, inclusive, a start\_lt\_index\_for\_rplr + N, exclusive. Los índices de largo plazo disponibles para imágenes de referencia inter-vistas pueden asignarse en el orden de view\_id, en el orden de las cámaras, o en el orden en el que se listan las dependencias de vistas en la extensión del conjunto de parámetros de secuencias para MVC. Los comandos de RPLR (sintaxis y semántica) permanecen sin cambio comparado con el estándar H.264/AVC.

30 Para el procesamiento temporal de relación directa, por ejemplo, para escalamiento de vector de movimiento, si ambas imágenes de referencia son imágenes inter-predicción (predicción intra-vistas), (esto es, las imágenes de referencia no están marcadas como "usadas para referencia inter-vistas"), entonces se sigue el proceso de decodificación de AVC. Si una de las dos imágenes de referencia es una imagen de inter-predicción y la otra es una imagen de predicción inter-vistas, la imagen de predicción inter-vistas es tratada como una imagen de referencia de largo plazo. De lo contrario (si ambas imágenes de referencia son imágenes inter-vistas), los valores view\_id o de indicadores del orden de las cámaras se usan en lugar de los valores de POC para el escalamiento del vector de movimiento.

40 Para la derivación de las ponderaciones de predicción para predicción ponderada implícita, se realiza el siguiente proceso. Si ambas imágenes de referencia son imágenes de inter-predicción (predicción inter-vistas) (es decir, no están marcadas como "usadas para referencia inter-vistas"), se sigue el proceso de decodificación de AVC. Si una de las dos imágenes de referencia es una imagen de inter-predicción y la otra es una imagen de predicción inter-vistas, entonces la imagen de predicción inter-vistas es tratada como una imagen de referencia de largo plazo. De lo contrario (es decir, ambas imágenes son imágenes de predicción inter-vistas), los valores de view\_id o de indicadores del orden de las cámaras se usan en lugar de los valores de POC para la derivación de los parámetros de predicción ponderados.

50 La presente invención está descrita en el contexto general de etapas de procedimiento, el cual puede implementarse en una realización por un producto de programa que incluya instrucciones ejecutables por ordenador, como un código de programa, implementado en un medio de lectura por ordenador y ejecutado por ordenadores en entornos de red. Ejemplos de medios de lectura por ordenador pueden incluir varios tipos de medios de almacenamiento incluyendo, pero sin limitarse a, unidades de memoria de dispositivos electrónicos, memoria de acceso aleatorio (RAM), memoria de solo lectura (ROM), discos compactos (CDs), discos versátiles digitales (DVDs) y otros dispositivos de almacenamiento interno o externo. Por lo general, los módulos de programa incluyen rutinas, programas, objetos, componentes, estructuras de datos, etc., que realizan tareas particulares o implementan tipos de datos abstractos particulares. Las instrucciones ejecutables por ordenador, las estructuras de datos asociados, y los módulos de programa representan ejemplos de código de programa para ejecutar etapas de los procedimientos descritos en el presente documento. La secuencia particular de tales instrucciones ejecutables o estructuras de datos asociados representan ejemplos de actos correspondientes para implementar las funciones descritas en tales etapas.

60 Las implementaciones de software de la presente invención podrían lograrse con técnicas de programación estándar con lógica basada en reglas y otra lógica para lograr las varias etapas de búsqueda de bases de datos, etapas de correlación, etapas de comparación y etapas de decisión. También se apreciará que las palabras "componente" y "modulo", como se emplean aquí y en las reivindicaciones, pretenden abarcar implementaciones que usan una o más líneas de código de software, y/o implementaciones de hardware, y/o equipo para la recepción de entrada de datos manuales.

5 La descripción anterior de realizaciones de la presente invención se ha presentado con propósitos de ilustración y de descripción. No se pretende que sean exhaustivas o que limiten la presente invención a la forma precisa descrita, y son posibles modificaciones y variaciones en vista de las enseñanzas anteriores o pueden adquirirse a partir de la práctica de la presente invención. Las realizaciones fueron elegidas y descritas con el fin de explicar los principios de la presente invención y su aplicación práctica para permitir a la persona experta en la técnica el uso de la presente invención en varias realizaciones y con varias modificaciones para ajustarse al uso particular contemplado.

**REIVINDICACIONES**

5 1. Un procedimiento de codificación de una pluralidad de vistas de una escena en un flujo de bits de vídeo codificado, conteniendo cada vista de dicha pluralidad de vistas una pluralidad de imágenes, comprendiendo el procedimiento:

10 proporcionar un elemento de señalización para cada imagen de una vista, indicando el elemento de señalización si la imagen correspondiente de dicha vista se utiliza o no como referencia para cualquier otra imagen que pertenece a una vista diferente, en donde el elemento de señalización es una señal y se  
10 señala en una encabezado de unidad de capa de abstracción de red de una unidad de capa de abstracción de red que contiene datos de vídeo codificados de dicha imagen correspondiente de dicha vista.

15 2. Un procedimiento de decodificación de un flujo de bits de vídeo codificado, una representación codificada de una pluralidad de vistas de una escena, conteniendo cada vista de dicha pluralidad de vistas una pluralidad de imágenes, comprendiendo el procedimiento:

20 recuperar un elemento de señalización para cada imagen de una vista desde el flujo de bits de vídeo codificado, indicando el elemento de señalización si la imagen correspondiente de dicha vista se utiliza o no como referencia para cualquier otra imagen que pertenece a una vista diferente, en donde el elemento de señalización es una señal y se recupera de un encabezado de unidad de capa de abstracción de red de una unidad de capa de abstracción de red que contiene datos de vídeo codificados de dicha imagen correspondiente de dicha vista.

25 3. Un procedimiento de acuerdo con la reivindicación 2, comprendiendo además el procedimiento:

30 si el elemento de señalización indica que la imagen de la vista no se utiliza como una referencia para cualquier otra imagen que pertenece a una vista diferente y si la imagen no se utiliza como referencia para cualquier otra imagen que pertenece a la misma vista, omitir la transmisión de una parte de la corriente de bits codificada correspondiente a la imagen.

4. Un procedimiento de acuerdo con la reivindicación 2, comprendiendo además el procedimiento:

35 si el elemento de señalización indica que la imagen de la vista no se utiliza como referencia para cualquier otra imagen que pertenece a una vista diferente y si la imagen no se utiliza como referencia para cualquier otra imagen que pertenece a la misma vista, omitir la decodificación de una parte de la corriente de bits codificada correspondiente a la imagen.

40 5. Un aparato para codificar una pluralidad de vistas de una escena en un flujo de bits de vídeo codificado, conteniendo cada vista de dicha pluralidad de vistas una pluralidad de imágenes, comprendiendo el aparato:

45 medios para proporcionar un elemento de señalización para cada imagen de una vista, indicando el elemento de señalización si la imagen correspondiente de dicha vista se utiliza o no como referencia para cualquier otra imagen que pertenece a una vista diferente, en donde el elemento de señalización es una señal y se  
45 señala en una encabezado de unidad de capa de abstracción de red de una unidad de capa de abstracción de red que contiene datos de vídeo codificados de dicha imagen correspondiente de dicha vista.

50 6. Un aparato para decodificar un flujo de bits de vídeo codificado, una representación codificada de una pluralidad de vistas de una escena, conteniendo cada vista de dicha pluralidad de vistas una pluralidad de imágenes, comprendiendo el aparato:

55 medios para recuperar un elemento de señalización para cada imagen de una vista desde el flujo de bits de vídeo codificado, indicando el elemento de señalización si la imagen correspondiente de dicha vista se utiliza o no como referencia para cualquier otra imagen que pertenece a una vista diferente, en el que el elemento de señalización es una señal y se recupera de un encabezado de unidad de capa de abstracción de red de una unidad de capa de abstracción de red que contiene datos de vídeo codificados de dicha imagen correspondiente de dicha vista.

60 7. Un aparato de acuerdo con la reivindicación 6, comprendiendo además el aparato:

65 medios para omitir la transmisión de una parte de la corriente de bits codificada correspondiente a la imagen si el elemento de señalización indica que la imagen de la vista no se utiliza como una referencia para cualquier otra imagen que pertenece a una vista diferente y si la imagen no se utiliza como referencia para cualquier otra imagen que pertenece a la misma vista.

8. Un aparato de acuerdo con la reivindicación 6, comprendiendo además el aparato:

- 5 medios para omitir la decodificación de una parte de la corriente de bits codificada correspondiente a la imagen si el elemento de señalización indica que la imagen de la vista no se utiliza como referencia para cualquier otra imagen que pertenece a una vista diferente y si la imagen no se utiliza como una referencia para cualquier otra imagen que pertenece a la misma vista.

FIG. 1

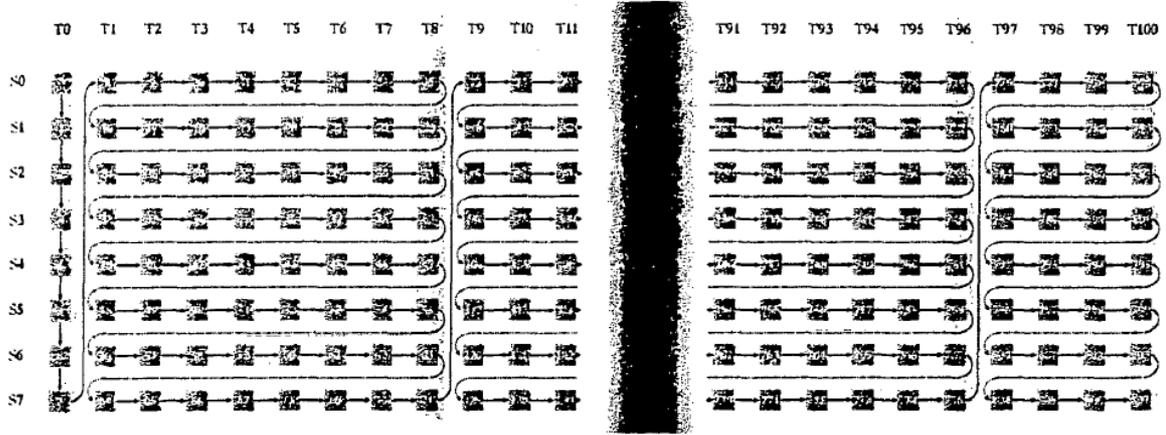


FIG. 2

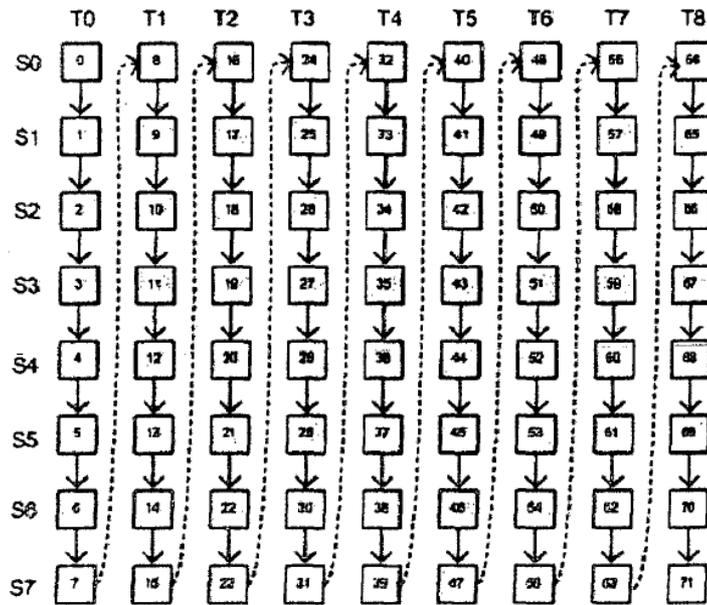
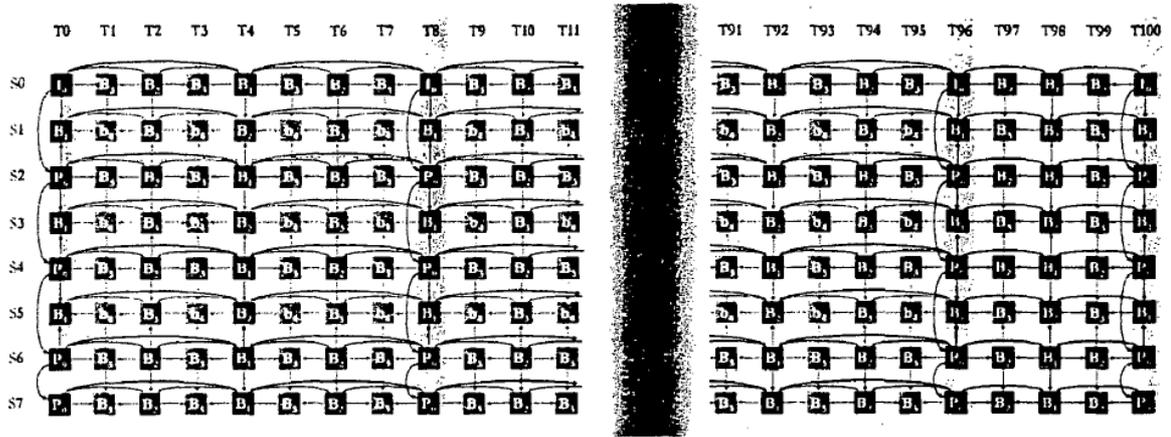


FIG. 3



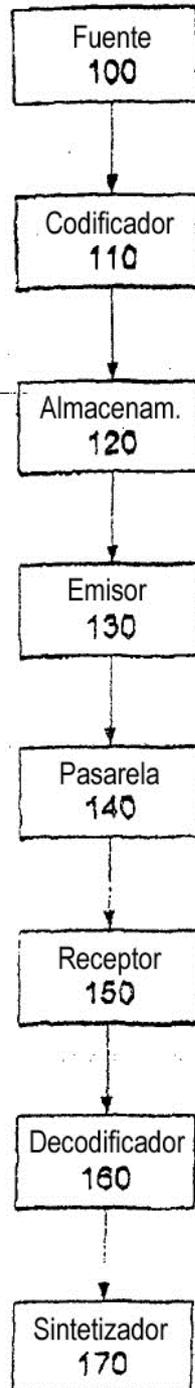


FIG. 4

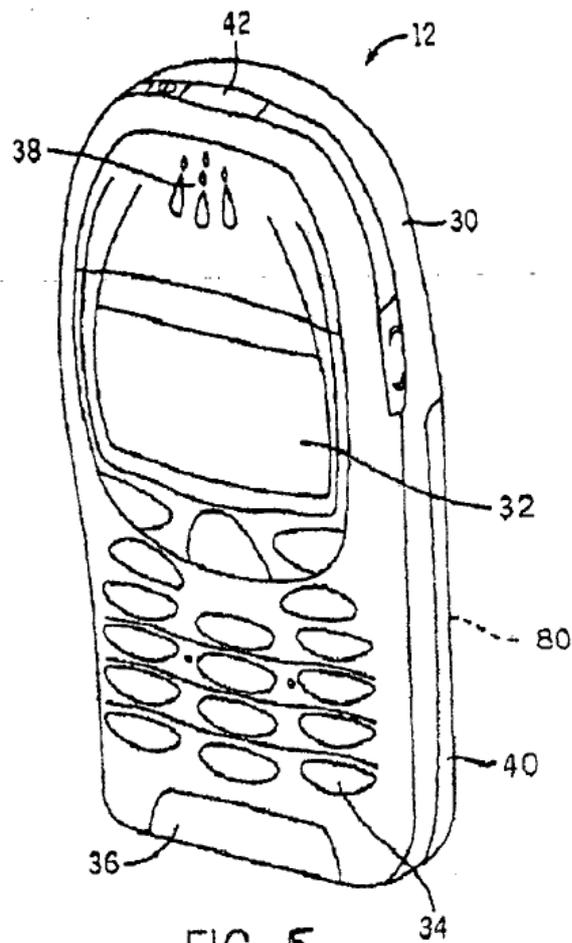


FIG. 5

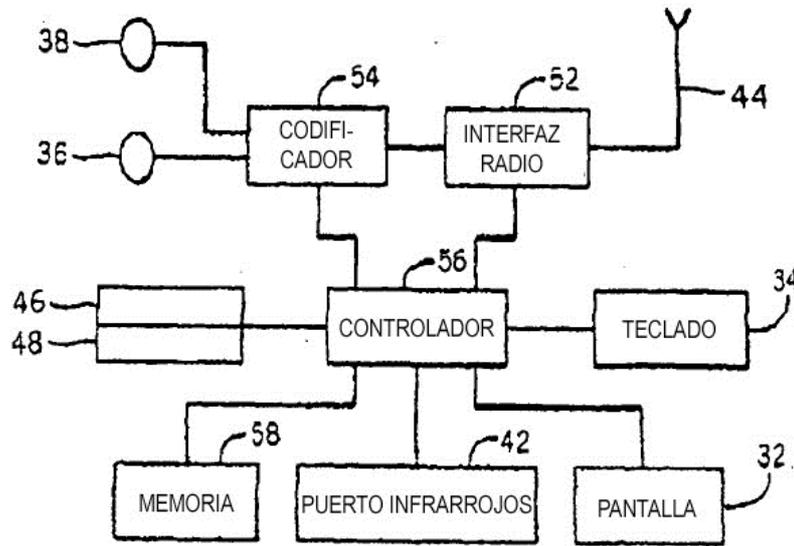


FIG. 6