



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

H.263

(03/96)

TRANSMISIÓN DE SEÑALES NO TELEFÓNICAS

**CODIFICACIÓN DE VÍDEO PARA
COMUNICACIÓN A BAJA
VELOCIDAD BINARIA**

Recomendación UIT-T H.263

(Anteriormente «Recomendación del CCITT»)

PREFACIO

El UIT-T (Sector de Normalización de las Telecomunicaciones) es un órgano permanente de la Unión Internacional de Telecomunicaciones (UIT). Este órgano estudia los aspectos técnicos, de explotación y tarifarios y publica Recomendaciones sobre los mismos, con miras a la normalización de las telecomunicaciones en el plano mundial.

La Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), que se celebra cada cuatro años, establece los temas que han de estudiar las Comisiones de Estudio del UIT-T, que a su vez producen Recomendaciones sobre dichos temas.

La aprobación de Recomendaciones por los Miembros del UIT-T es el objeto del procedimiento establecido en la Resolución N.º 1 de la CMNT (Helsinki, 1 al 12 de marzo de 1993).

La Recomendación UIT-T H.263 ha sido preparada por la Comisión de Estudio 15 (1993-1996) del UIT-T y fue aprobada por el procedimiento de la Resolución N.º 1 de la CMNT el 19 de marzo de 1996.

NOTA

En esta Recomendación, la expresión «Administración» se utiliza para designar, en forma abreviada, tanto una administración de telecomunicaciones como una empresa de explotación reconocida de telecomunicaciones.

© UIT 1996

Es propiedad. Ninguna parte de esta publicación puede reproducirse o utilizarse, de ninguna forma o por ningún medio, sea éste electrónico o mecánico, de fotocopia o de microfilm, sin previa autorización escrita por parte de la UIT.

ÍNDICE

		<i>Página</i>
1	Alcance.....	1
2	Referencias.....	1
3	Especificación breve	1
	3.1 Entrada y salida vídeo.....	1
	3.2 Entrada y salida digital	1
	3.3 Frecuencia de muestreo	1
	3.4 Algoritmo de codificación de fuente	2
	3.5 Velocidad binaria.....	3
	3.6 Almacenamiento en tampón	3
	3.7 Simetría de la transmisión.....	3
	3.8 Tratamiento de los errores	3
	3.9 Funcionamiento multipunto.....	4
4	Codificador de fuente.....	4
	4.1 Formato de fuente.....	4
	4.2 Algoritmo de codificación de fuente vídeo.....	5
	4.3 Control de la codificación.....	8
	4.4 Actualización forzada	8
	4.5 Alineación en byte de los códigos de comienzo	8
5	Sintaxis y semántica	8
	5.1 Capa de imagen.....	9
	5.2 Capa de grupo de bloques.....	12
	5.3 Capa de macrobloque	13
	5.4 Capa de bloque	18
6	Proceso de decodificación.....	18
	6.1 Compensación de movimiento.....	18
	6.2 Coeficientes de decodificación	24
	6.3 Reconstrucción de los bloques.....	25
	Anexo A – Especificación de la exactitud de la transformada inversa.....	26
	Anexo B – Decodificador ficticio de referencia.....	27
	Anexo C – Consideraciones sobre el funcionamiento multipunto	28
	C.1 Petición de imagen congelada.....	28
	C.2 Petición de actualización rápida	28
	C.3 Liberación de imagen congelada	28
	C.4 Multipunto de presencia continua (CPM, continuous presence multipoint) (no se utiliza para la Recomendación H.324)	29
	Anexo D – Modo vector de movimiento sin restricción.....	29
	D.1 Vectores de movimiento en fronteras de imagen.....	29
	D.2 Extensión de la gama de vectores de movimiento	30
	Anexo E – Modo codificación aritmética basada en sintaxis	30
	E.1 Introducción.....	30
	E.2 Especificación del codificador SAC.....	31
	E.3 Especificación del decodificador SAC	31
	E.4 Sintaxis	32

	<i>Página</i>
E.5 PSC_FIFO	33
E.6 Símbolos de longitud fija	33
E.7 Símbolo de longitud no fija	33
E.8 Modelos SAC	34
Anexo F – Modo predicción avanzada	36
F.1 Introducción	36
F.2 Cuatro vectores de movimiento por macrobloque	36
F.3 Compensación de movimiento para bloques de luminancia superpuestos	38
Anexo G – Modo trama PB	40
G.1 Introducción	40
G.2 Tramas PB y bloques INTRA	40
G.3 Capa de bloque	41
G.4 Cálculo de los vectores de imagen B y con trama PB	41
G.5 Predicción de un bloque B en una trama PB	41
Anexo H – Corrección de errores hacia adelante para señales de vídeo codificadas	43
H.1 Introducción	43
H.2 Alineación de trama para la corrección de errores	43
H.3 Código de corrección de errores	43
H.4 Plazo de reenganche para la alineación del corrector de errores	44

RESUMEN

Esta Recomendación especifica una representación codificada, que se puede usar para comprimir el componente de imagen en movimiento de los servicios audiovisuales a velocidades binarias bajas. La configuración básica del algoritmo de codificación de fuente vídeo se basa en la Recomendación H.261, y es un híbrido de la predicción entre imágenes que utiliza la redundancia temporal y la codificación con transformada de la señal restante para reducir la redundancia espacial. El codificador de fuente puede funcionar con cinco formatos de imagen normalizados: sub-QCIF, QCIF, CIF, 4CIF y 16CIF.

El decodificador tiene capacidad de compensación de movimiento, lo que permite la incorporación opcional de esta técnica en el codificador. En la compensación de movimiento se utiliza la precisión de mitad de pixel, mientras que en la Recomendación H.261 se utiliza la precisión de pixel entero y un filtro de bucle. Se emplea la codificación de longitud variable para la transmisión de los símbolos.

Además del algoritmo de codificación de fuente vídeo básico, se incluyen cuatro opciones de codificación negociables para mejorar el funcionamiento: vector de movimiento sin restricción, codificación aritmética basada en sintaxis, predicción avanzada y trama PB. Todas estas opciones se pueden emplear juntas o por separado.

CODIFICACIÓN DE VÍDEO PARA COMUNICACIÓN A BAJA VELOCIDAD BINARIA

(Ginebra, 1996)

1 Alcance

Esta Recomendación especifica una representación codificada que se puede usar para comprimir el componente de imagen en movimiento de los servicios audiovisuales a velocidades binarias bajas. La configuración básica del algoritmo de codificación de fuente vídeo se basa en la Recomendación H.261. Se incluyen cuatro opciones de codificación negociables para mejorar el funcionamiento.

2 Referencias

Las Recomendaciones y demás referencias siguientes contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y demás referencias son objeto de revisiones, por lo que se preconiza que todos los usuarios de la presente Recomendación investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y demás referencias citadas a continuación. Se publica regularmente una lista de las Recomendaciones UIT-T actualmente vigentes.

- [1] Recomendación UIT-T H.223 (1996), *Protocolo de multiplexación para comunicación multimedias a baja velocidad binaria*.
- [2] Recomendación UIT-T H.242 (1996), *Sistemas para el establecimiento de comunicaciones entre terminales audiovisuales con utilización de canales digitales de hasta 2 Mbit/s*.
- [3] Recomendación UIT-T H.245 (1996), *Protocolo de control para comunicación multimedios*.
- [4] Recomendación UIT-T H.261 (1993), *Códec vídeo para servicios audiovisuales a $p \times 64$ kbit/s*.
- [5] Rec. UIT-T H.262 (1995) | ISO/CEI 13818-2:1995, *Tecnología de la información – Codificación genérica de imágenes en movimiento e información de audio asociada: Vídeo*.
- [6] Recomendación UIT-T H.320 (1996), *Sistemas y equipos terminales videotelefónicos de banda estrecha*.
- [7] Recomendación UIT-T H.324 (1996), *Terminal para comunicación multimedias a baja velocidad binaria*.

3 Especificación breve

En la Figura 1 aparece un diagrama de bloques resumido del códec.

3.1 Entrada y salida vídeo

Para poder abarcar con una sola Recomendación la utilización dentro y entre regiones que emplean normas de televisión de 625 y 525 líneas, el codificador de fuente actúa sobre imágenes basadas en un formato intermedio común (CIF, *common intermediate format*). Las normas de las señales de televisión de entrada y salida, que pueden, por ejemplo, ser compuestas o de componentes analógicas o digitales, no son objeto de Recomendaciones, como tampoco lo son los métodos para realizar cualquier conversión necesaria de y hacia el formato de codificación de fuente.

3.2 Entrada y salida digital

El codificador vídeo proporciona un tren binario digital autocontenido que se puede combinar con otras señales multifacilidades (tal como se define en la Recomendación H.223, por ejemplo). El decodificador vídeo efectúa el proceso inverso.

3.3 Frecuencia de muestreo

Las imágenes se muestrean a un múltiplo entero de la frecuencia de línea vídeo. Este reloj de muestreo y el reloj de red digital son asíncronos.

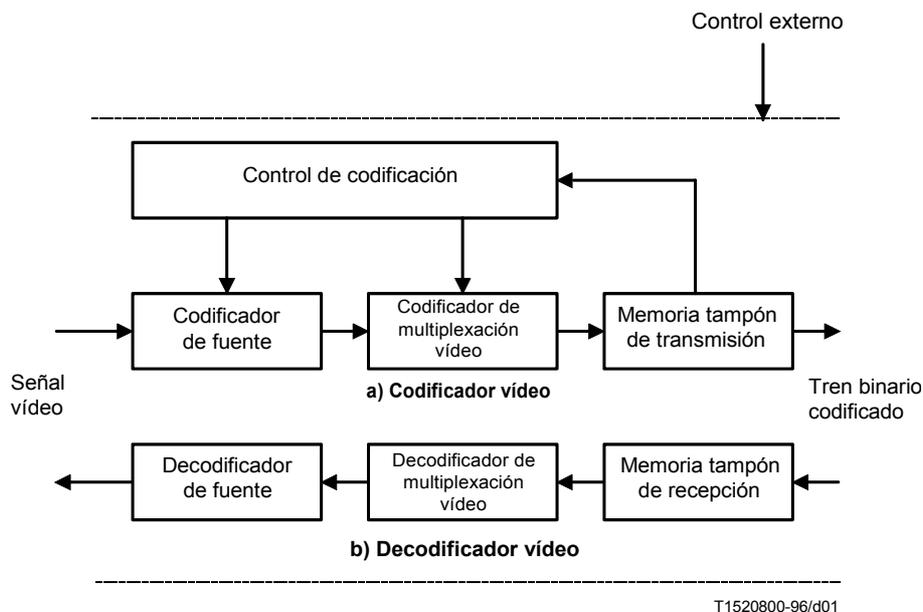


FIGURA 1/H.263
Diagrama de bloques resumido del códec vídeo

3.4 Algoritmo de codificación de fuente

Se adopta un híbrido de predicción entre imágenes para utilizar redundancia temporal y codificación de la transformada de la señal restante para reducir la redundancia espacial. El decodificador tiene capacidad de compensación de movimiento, lo que permite la incorporación opcional de esta técnica en el codificador. Se utiliza una precisión de mitad de pixel para la compensación de movimiento, en vez de la precisión de pixel entero y filtro de bucle de la Recomendación H.261. En la transmisión de los símbolos se emplea la codificación de longitud variable.

Además del algoritmo de codificación básico H.263, se incluyen cuatro opciones de codificación negociables, que se describirán en las subcláusulas siguientes. Estas opciones se pueden utilizar juntas o por separado.

3.4.1 Modo vector de movimiento sin restricción

En este modo opcional, los vectores de movimiento están autorizados a apuntar fuera de la imagen. Los pixels de borde se utilizan como predicción de los pixels «no existentes». Con este modo se consigue una ganancia significativa cuando hay movimiento a través de los bordes de la imagen, especialmente con los formatos de imagen más pequeños (véase también el Anexo D). Además, este modo incluye una extensión de la gama de vectores de movimiento de modo que se puedan utilizar vectores de movimiento mayores, lo que es especialmente útil en el caso de movimiento de la cámara.

3.4.2 Modo de codificación aritmética basada en sintaxis

En este modo facultativo, se utiliza la codificación aritmética en vez de la codificación de longitud variable. La relación señal/ruido (SNR) y las imágenes reconstruidas serán las mismas, pero se producirá un número considerablemente menor de bits (véase también el Anexo E).

3.4.3 Modo predicción avanzada

En este modo opcional, se utiliza la compensación de movimiento bloques superpuestos (OBMC, *overlapped block motion compensation*) para la parte de luminancia de las imágenes P (véase también el Anexo F). En algunos macrobloques de la imagen, se utilizan cuatro vectores de 8×8 en vez de un vector de 16×16 . El codificador decidirá el tipo de vector que utilizará. Cuatro vectores utilizan más bits, pero dan una predicción mejor. La aplicación de este modo permite, en general, un perfeccionamiento considerable. Más especialmente, se consigue una ganancia subjetiva porque OBMC resulta en menos artefactos de bloqueo.

3.4.4 Modo trama PB

Una trama PB consiste en dos imágenes que se codifican como una unidad. El nombre «PB» proviene del nombre de los tipos de imagen de la Recomendación H.262, en la que hay imágenes P e imágenes B. Por consiguiente, una trama PB consiste en una imagen P, que se predice a partir de la anterior imagen P decodificada, y una imagen B, que se predice a partir de la anterior imagen P decodificada y la imagen P que está siendo decodificada en ese momento. Se eligió la denominación imagen B porque partes de imágenes B se pueden predecir bidireccionalmente a partir de imágenes P pasadas y futuras. Con esta opción de codificación se consigue aumentar considerablemente el periodo de transmisión de la imagen sin aumentar demasiado la velocidad binaria.

3.5 Velocidad binaria

El reloj de transmisión se proporciona por medios externos. La velocidad binaria de vídeo puede ser variable. En esta Recomendación no se imponen restricciones a la velocidad binaria de vídeo; estas restricciones las determinará el terminal o la red.

3.6 Almacenamiento en tampón

El codificador controlará su tren binario de salida para cumplir los requisitos del decodificador ficticio de referencia definido en el Anexo B. Los datos de vídeo se proporcionarán en cada ciclo de reloj válido. Esto se consigue mediante relleno de tipo de macrobloque y patrón de grupo codificado (MCBCP, *macroblock type and coded block pattern*) (véanse los Cuadros 4 y 5) o, cuando se utiliza corrección de errores hacia adelante, también mediante tramas de relleno de corrección de errores hacia adelante (véase el Anexo H).

El número de bits creado por la codificación de cualquier imagen no excederá de un valor máximo especificado por el parámetro BPPmaxKb que se mide en unidades de 1024 bits. El valor admisible mínimo del parámetro BPPmaxKb depende del formato de la imagen fuente más grande que ha sido negociado para su utilización en el tren de bits (véase el Cuadro 1). Un codificador puede utilizar un valor mayor para BPPmaxKb que el especificado en el Cuadro 1 a condición de que este valor mayor sea negociado primero por medios externos, por ejemplo, aplicando la Recomendación H.245.

CUADRO 1/H.263

Parámetro BPPmaxKb para cada uno de los formatos de imagen fuente

Formato de la fuente	BPPmaxKb
sub-QCIF	64
QCIF	64
CIF	256
4CIF	512
16CIF	1024

3.7 Simetría de la transmisión

El códec se puede utilizar para la comunicación visual bidireccional o unidireccional.

3.8 Tratamiento de los errores

El tratamiento de los errores se suministra por medios externos (por ejemplo, Recomendación H.223). Si no se proporciona por medios externos (por ejemplo, la Recomendación H.221), se puede utilizar la alineación de trama y la corrección de errores opcional, como se describe en el Anexo H.

El decodificador puede enviar una instrucción para codificar uno o más GOB de la imagen próxima en modo INTRA con los parámetros de codificación adecuados para evitar el desbordamiento de la memoria tampón. El decodificador también puede enviar una instrucción de que se transmita únicamente los encabezamientos GOB no vacíos. Estas señales se transmiten por medios externos (por ejemplo, Recomendación H.245).

3.9 Funcionamiento multipunto

En el Anexo C se describen las facilidades necesarias para el funcionamiento multipunto con conmutación.

4 Codificador de fuente

4.1 Formato de fuente

El codificador de fuente trabaja con imágenes no entrelazadas que se presentan 30 000/1001 (aproximadamente 29,97) veces por segundo. La tolerancia de la frecuencia de imagen es de ± 50 ppm.

Las imágenes se codifican como una componente de luminancia y dos componentes de diferencia de color (Y , C_R y C_B). Estas componentes y los códigos que representan sus valores muestreados son los que define la Recomendación 601 del UIT-R.

- Negro = 16
- Blanco = 235
- Diferencia de color nula = 128
- Diferencia de color máxima = 16 y 240.

Estos valores son nominales y el algoritmo de codificación funciona con valores de entrada comprendidos entre 1 y 254.

Hay cinco formatos de imagen normalizados: sub-QCIF, QCIF, CIF, 4CIF y 16CIF. Para cada uno de estos formatos de imagen, la estructura de muestreo de luminancia es dx pixels por línea y dy líneas por imagen en una disposición ortogonal. El muestreo de cada una de las dos componentes de diferencia de color se efectúa a $dx/2$ pixels por línea y $dy/2$ líneas por imagen, ortogonal. Los valores de dx , dy , $dx/2$ y $dy/2$ se ofrecen en el Cuadro 2 para cada uno de los formatos de imagen.

CUADRO 2/H.263

Número de pixels por línea y número de líneas de cada formato de imagen de H.263

Formato de imagen	Número de pixels de luminancia (dx)	Número de líneas de luminancia (dy)	Número de pixels de crominancia (dx/2)	Número de líneas de crominancia (dy/2)
sub-QCIF	128	96	64	48
QCIF	176	144	88	72
CIF	352	288	176	144
4CIF	704	576	352	288
16CIF	1408	1152	704	576

En cada uno de los formatos de imagen, las muestras de diferencia de color se colocan de forma tal que sus límites de bloque coincidan con los límites de bloque de luminancia, como se muestra en la Figura 2. La relación de aspecto de pixel es la misma para cada uno de estos formatos de imagen, y es la misma que se define para QCIF y CIF en la Recomendación H.261: $(4/3) * (288/352)$. El área de la imagen cubierta por todos los formatos de imagen, salvo el sub-QCIF, tiene una relación de aspecto 4:3.

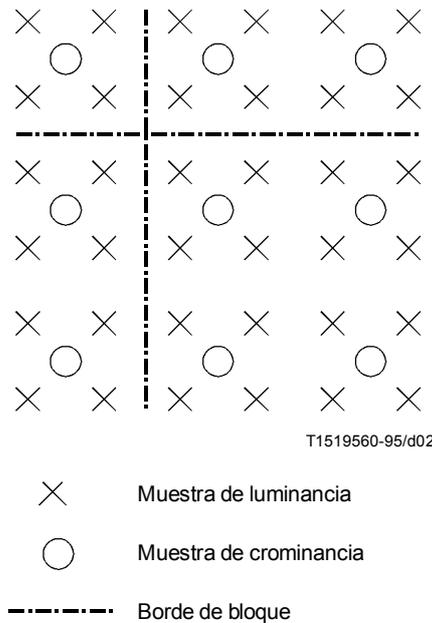


FIGURA 2/H.263
Posición de las muestras de luminancia y crominancia

Todos los decodificadores deberán poder funcionar con sub-QCIF y también con QCIF. Algunos decodificadores también podrán funcionar con CIF, 4CIF o 16CIF. Los codificadores deberán ser capaces de funcionar con uno de los formatos sub-QCIF y QCIF. Los codificadores determinarán cuál de esos dos formatos se utilizará, y no están obligados a funcionar con ambos. Algunos codificadores también pueden funcionar con CIF, 4CIF o 16CIF. Los medios externos, como la Recomendación H.245, indican los formatos que puede manipular el decodificador. Para un panorama completo de los posibles formatos de imagen y los algoritmos de codificación vídeo, se remite al lector a la descripción del terminal, por ejemplo, la Recomendación H.324.

NOTA – En el caso de CIF, el número de pixels por línea es compatible con el muestreo de las porciones activas de las señales de luminancia y de diferencia de color de fuentes de 525 ó 625 líneas a 6,75 y 3,375 MHz, respectivamente. Estas frecuencias tienen una relación simple con las de la Recomendación 601 del UIT-R.

Se proporcionarán medios para restringir el periodo máximo de transmisión de imagen de los codificadores colocando un número mínimo de imágenes no transmitidas entre las imágenes transmitidas. La selección de esa cantidad mínima se hará por medios externos (por ejemplo, Recomendación H.245). En el cálculo de la cantidad mínima de imágenes no transmitidas en el modo trama PB, las imágenes P y B de una unidad de tramas PB se tratan como dos imágenes separadas.

4.2 Algoritmo de codificación de fuente vídeo

El codificador de fuente se muestra de forma generalizada en la Figura 3. Los principales elementos son la predicción, la transformación de bloques y la cuantificación.

4.2.1 Grupos de bloques, macrobloques y bloques

Cada imagen se divide en grupos de bloques (GOB, *group of blocks*). Un grupo de bloques (GOB) comprende $k * 16$ líneas, según el formato de imagen ($k = 1$ para sub-QCIF, QCIF y CIF; $k = 2$ para 4CIF; $k = 4$ para 16CIF). El número de GOB por imagen es 6 para sub-QCIF, 9 para QCIF y 18 para CIF, 4CIF y 16CIF. La numeración de GOB se efectúa mediante el barrido vertical de los GOB, comenzando por el GOB superior (número 0) y terminando con el GOB inferior. En la Figura 4 se da un ejemplo de la disposición de los GOB en una imagen de formato CIF. Los datos de cada GOB consisten en un encabezamiento GOB (que puede estar vacío), seguido de datos de macrobloque. Los datos de GOB se transmiten por GOB en orden de GOB creciente.

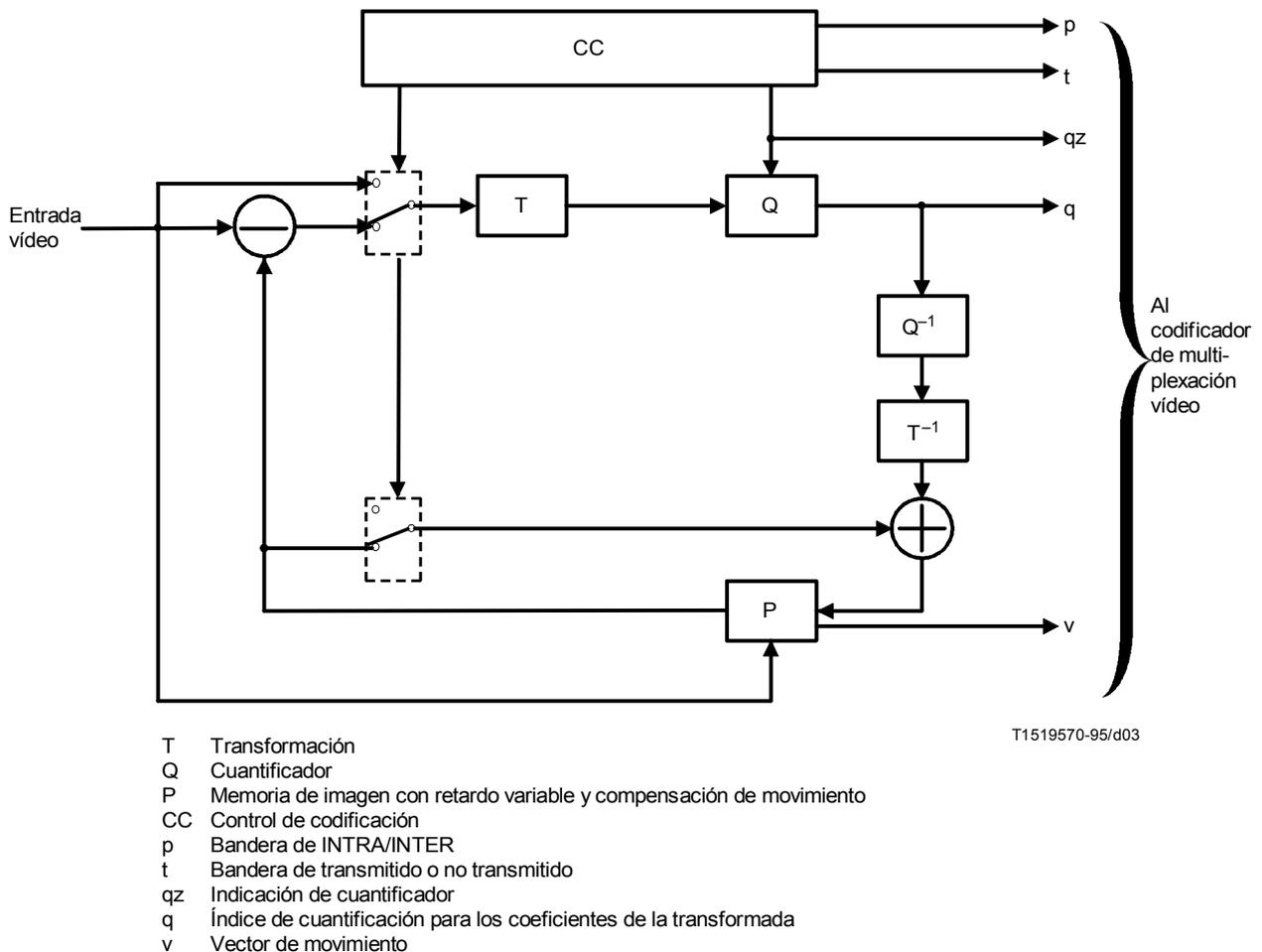


FIGURA 3/H.263
Codificador de fuente

Cada GOB se divide en macrobloques. Un macrobloque se relaciona con 16 pixels por 16 líneas de Y, y con 8 pixels por 8 líneas correspondientes espacialmente de C_B y C_R. Además, un macrobloque consiste en cuatro bloques de luminancia y los dos bloques de diferencia de color correspondientes espacialmente, como se muestra en la Figura 5. Cada bloque de luminancia o de crominancia se relaciona con 8 pixels por 8 líneas de Y, C_B o C_R. Un GOB comprende una fila de macrobloques para sub-QCIF, QCIF y CIF, dos filas de macrobloques para 4CIF y cuatro filas de macrobloques para 16CIF.

La numeración de macrobloque se efectúa mediante un barrido horizontal de las filas de macrobloque de izquierda a derecha, comenzando por la línea de macrobloque superior y terminando por la fila de macrobloque más baja. Los datos para los macrobloques se transmiten por macrobloque aumentando el número de macrobloque. Los datos para los bloques se transmiten por bloque al aumentar el número de bloque (véase la Figura 5).

Los criterios de elección del modo y la transmisión de un bloque no son objeto de recomendación y pueden variar dinámicamente como parte de la estrategia de control de la codificación. Los bloques transmitidos se transforman y los coeficientes resultantes se cuantifican y se codifican en entropía.

4.2.2 Predicción

La predicción se efectúa entre imágenes y se puede aumentar por compensación de movimiento (véase 4.2.3). El modo de codificación en el que se aplica la predicción se denomina INTER; el modo de codificación se denomina INTRA cuando no se aplica la predicción. El modo de codificación INTRA se puede señalar en el nivel de la imagen (INTRA para imágenes I o INTER para imágenes P) o en el nivel de macrobloque en las imágenes P. En el modo trama PB opcional, las imágenes B están siempre codificadas en modo INTER. Las imágenes B se predicen parcialmente en forma bidireccional (véase el Anexo G).

					0					
					1					
					2					
					3					
					4					
					5					
					6					
					7					
					8					
					9					
					10					
					11					
					12					
					13					
					14					
					15					
					16					
					17					

FIGURA 4/H.263

Disposición de grupo de bloque en una imagen CIF

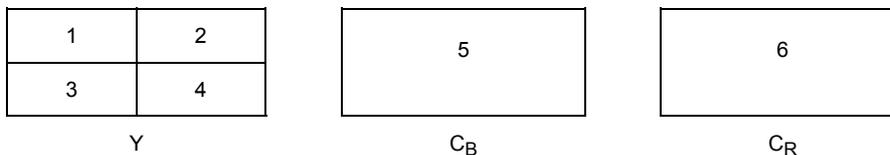


FIGURA 5/H.263

Disposición de los bloques en un macrobloque

4.2.3 Compensación de movimiento

El decodificador aceptará un vector por macrobloque o, si se utiliza el modo predicción avanzada, aceptará uno o cuatro vectores por macrobloque (véase el Anexo F). Cuando se utiliza el modo trama PB, se puede transmitir un vector delta adicional por macrobloque para la adaptación de vectores de movimiento destinado a la predicción de macrobloque B.

Los componentes horizontal y vertical de los vectores de movimiento tienen valores de un entero o de mitad de entero. En el modo de predicción por defecto, estos valores están restringidos a la gama $[-16, 15,5]$ que es también válida para los componentes de vectores de movimiento hacia adelante y hacia atrás de las imágenes B. Sin embargo, en el modo vector de movimientos sin restricción, la gama máxima para los componentes de vectores es $[-31,5, 31,5]$, con la restricción que sólo se pueden alcanzar los valores que están dentro de una gama de $[-16, 15,5]$ alrededor del predictor para cada componente de vector de movimiento si el predictor está en la gama $[-15,5, 16]$. Si el predictor está fuera de la gama $[-15,5, 16]$, se pueden alcanzar todos los valores dentro de la gama $[-31,5, 31,5]$ con el mismo signo que el predictor más el valor cero (véase también el Anexo D).

Un valor positivo del componente horizontal o vertical del vector significa que la predicción se forma a partir de los pixels de la imagen referenciada que están espacialmente a la derecha o por debajo de los pixels que se predicen.

Los vectores de movimiento están restringidos de modo que todos los pixels referenciados por ellos se encuentren dentro de la zona de imagen codificada, salvo cuando se utiliza el modo de vector de movimiento sin restricción y/o el modo predicción avanzada (véanse los Anexos D y F).

4.2.4 Cuantificación

El número de cuantificadores es 1 para el primer coeficiente de bloque INTRA y 31 para todos los demás coeficientes. Dentro de un macrobloque, se utiliza el mismo cuantificador para todos los coeficientes, excepto el primero de los bloques INTRA. No se definen los niveles de decisión. El primer coeficiente de los bloques INTRA es, nominalmente, el valor de la transformada en continua (cc) cuantificado uniformemente con un paso de valor 8. Cada uno de los restantes 31 cuantificadores utilizan niveles de reconstrucción de espaciado igual con una zona muerta central alrededor de cero y un paso de valor par comprendido entre 2 y 62. Para las fórmulas exactas, véase 6.2.

NOTA – En el caso de tamaños más pequeños de pasos de cuantificación, no se puede representar la gama dinámica completa de los coeficientes de la transformada.

4.3 Control de la codificación

Se pueden cambiar varios parámetros para controlar la velocidad de generación de datos de vídeo codificados. Se trata del proceso efectuado antes del codificador de fuente, el cuantificador, el criterio de significación de bloque y el submuestreo temporal. Las proporciones de estas medidas en la estrategia de control global no son objeto de recomendación.

Al ser invocado, el submuestreo temporal se ejecuta mediante el descarte de imágenes completas.

Un decodificador puede indicar su preferencia por un compromiso específico entre la resolución espacial y la temporal de la señal vídeo. El codificador señalará su compromiso por defecto al comienzo de la llamada e indicará si es capaz de responder a las solicitudes del decodificador de cambiar ese compromiso. Estas señales se transmiten por medios externos (por ejemplo, Recomendación H.245).

4.4 Actualización forzada

Esta función se realiza forzando la utilización del modo INTRA del algoritmo de codificación. El esquema de actualización no se define. Para controlar la acumulación de errores por desajuste de la transformada inversa, cada macrobloque se codificará en modo INTRA al menos una vez cada 132 veces cuando se transmiten los coeficientes de este macrobloque en imágenes P.

4.5 Alineación en byte de los códigos de comienzo

La alineación en byte de los códigos de comienzo se consigue insertando una palabra de código de menos de 8 bits cero antes del código del comienzo, de manera que el primer bit del código de comienzo es el primer bit (más significativo) de un byte. Por consiguiente, un código de comienzo está alineado en byte si la posición de su bit más significativo es un múltiplo de 8 bits del primer bit del tren binario H.263. Es obligatorio que los códigos de comienzo de todos los tipos de imagen estén alineados en byte, y los códigos de comienzo de GOB y EOS no están sujetos a esa obligación.

NOTAS

- 1 El número de bits utilizado para una imagen determinada es variable pero siempre un múltiplo de 8 bits.
- 2 El procedimiento H.324 requiere codificadores de la Recomendación H.263 para alinear los códigos de comienzo de imagen con el comienzo de las unidades de información lógicas pasadas a la capa de adaptación (las SDU de AL).

5 Sintaxis y semántica

El multiplex de vídeo tiene una estructura jerárquica de cuatro capas. De arriba hacia abajo, esas capas son:

- imagen;
- grupo de bloques;
- macrobloque;
- bloque.

El diagrama de sintaxis aparece en la Figura 7. Las abreviaturas y la semántica se definen en las secciones siguientes.

A menos que se especifique otra cosa, el bit más significativo se transmite primero. Es el bit 1, situado en el extremo izquierdo de las tablas de bits de esta Recomendación. A menos que se especifique otra cosa, todos los bits no utilizados o de reserva se fijan a «1». Los bits de reserva no se utilizarán hasta que la UIT especifique sus funciones.

5.1 Capa de imagen

Los datos de cada imagen consisten en un encabezamiento de imagen, seguido de datos de grupo de datos, seguidos de un código de fin de secuencia y bits de relleno. La estructura aparece en la Figura 6. PSBI sólo está presente si así lo indica CPM. TR_B y DBQUANT sólo están presentes si PTYPE indica «trama PB». Las combinaciones de PSPARE y PEI pueden no estar presentes. EOS puede no estar presente, mientras que ESTUF puede estar presente sólo si EOS está presente. Los encabezamientos de imagen de las imágenes descartadas no se transmiten.

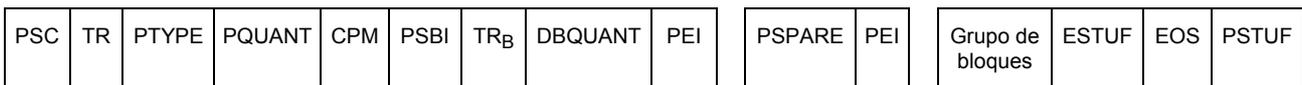


FIGURA 6/H.263

Estructura de la capa de imagen

5.1.1 Código de comienzo de imagen (PSC, *picture start code*) (22 bits)

PSC es una palabra de 22 bits. Su valor es 0000 0000 0000 0000 1 00000. Todos los códigos de comienzo de imagen estarán alineados en byte. Esto se consigue insertando PSTUF antes del código de comienzo, de manera que el primer bit del código de comienzo es el primer bit (más significativo) de un byte.

5.1.2 Referencia temporal (TR, *temporal reference*) (8 bits)

Número de 8 bits que tiene 256 valores posibles. Se forma incrementando en uno el valor que tenía en el encabezamiento de imagen transmitido previamente, más el número de imágenes no transmitidas (a 29,97 Hz) desde la transmitida anteriormente. La aritmética se ejecuta con sólo los ocho bits menos significativos (LSB, *least significant bit*). En el modo trama PB opcional, TR se refiere únicamente a las imágenes P; para la referencia temporal de las imágenes B, véase 5.1.7.

5.1.3 Tipo de información (PTYPE, *type information*) (13 bits)

Información sobre la imagen completa:

- Bit 1: Siempre «1», para evitar emulación del código de comienzo.
- Bit 2: Siempre «0» para distinción con la Recomendación H.261.
- Bit 3: Indicador de pantalla dividida, «0» desactivado, «1» activado.
- Bit 4: Indicador de cámara de documento, «0» desactivado, «1» activado.
- Bit 5: Liberación de imagen congelada, «0» desactivado, «1» activado.
- Bits 6-8: Formato de fuente «000», prohibido, «001» sub-QCIF, «010» QCIF, «011» CIF, «100» 4CIF, «101» 16CIF, «110» reservado, «111» reservado.
- Bit 9: Tipo de codificación de imagen, «0» INTRA, (imagen I), «1» INTER, (imagen P).
- Bit 10: Modo vector de movimiento sin restricción, opcional, «0» off, «1» on.
- Bit 11: Modo codificación aritmética basada en sintaxis, opcional, «0» off, «1» on.
- Bit 12: Modo predicción avanzada, opcional, «0» off, «1» on.
- Bit 13: Modo trama PB, opcional, «0» imagen I o P normal, «1» trama PB.

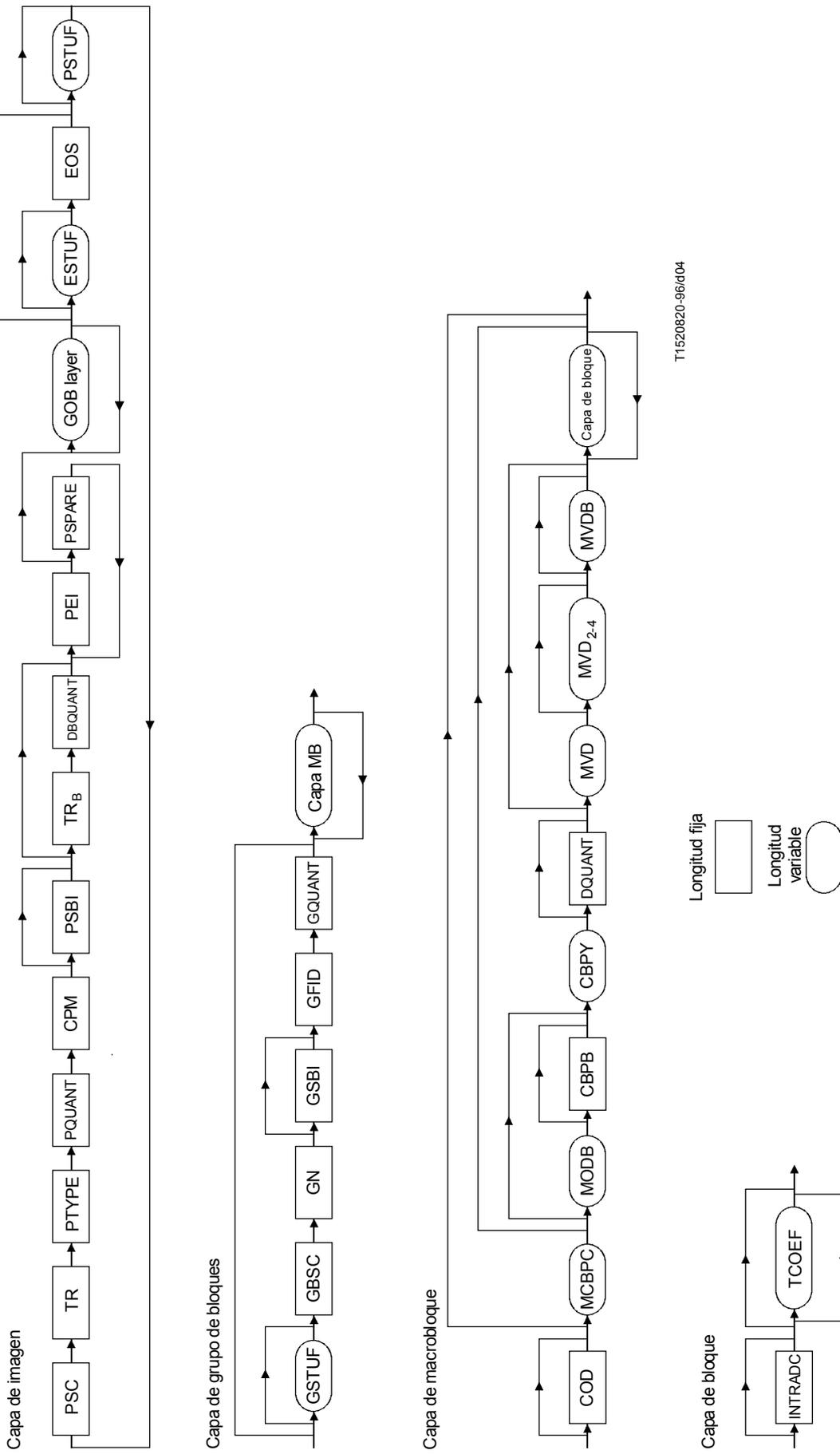


FIGURA 7/H.263

Documento de estándar del Área Técnica de Video

«Indicador de pantalla dividida» es una señal que indica que las mitades superior e inferior de la imagen decodificada se pueden visualizar una al lado de la otra. Este bit no tiene efectos directos sobre la codificación o decodificación de la imagen.

«Liberación de imagen congelada» es una señal procedente de un codificador, que responde a una solicitud de retransmisión de paquete (cuando no hay acuse de recibo) o a una petición de actualización rápida (véase también el Anexo C), y permite a un decodificador salir del modo imagen congelada y presentar la imagen decodificada de manera normal.

Si los bits 6 a 8 indican un formato de fuente diferente del encabezamiento de imagen anterior, la imagen vigente será una imagen I.

Los bits 10-13 se refieren a modos opcionales, que se utilizan únicamente después de una negociación entre el codificador y el decodificador (véanse también los Anexos D, E, F y G, respectivamente). Cuando el bit 9 se pone a «0», el bit 13 también se fija a «0».

5.1.4 Información de cuantificador (PQUANT, *quantizer information*) (5 bits)

Palabra de código de longitud fija de 5 bits, que indica el cuantificador QUANT que se utilizará para la imagen hasta que sea actualizada por cualquier GQUANT o DQUANT subsiguiente. Las palabras de código son la representación binaria natural de los valores de QUANT que, puesto que tienen un paso mitad, están comprendidos entre 1 y 31.

5.1.5 Multipunto de presencia continua (CPM, *continuous presence multipoint*) (1 bit)

Palabra de código de 1 bit que señala la utilización del modo multipunto de presencia continua facultativo (CPM); «0» off, «1» on. Para el uso de CPM, véase el Anexo C.

5.1.6 Indicador de subtren de bits de imagen (PSBI, *picture sub-bit stream indicator*) (2 bits)

Una palabra de código de longitud fija de 2 bits, que sólo está presente si el modo multipunto de presencia continua es indicado por CPM. Las palabras de código son la representación binaria natural del número del subtren de bits para el encabezamiento de la imagen y toda la información siguiente hasta la próxima imagen o código de comienzo GOB (véase también el Anexo C).

5.1.7 Referencia temporal para imágenes B (TR_B, *temporal reference for B-pictures*) (3 bits)

TR_B está presente si PTYPE indica «trama PB» (véase también el Anexo G), e indica el número de imágenes no transmitidas (a 29,97 Hz) desde la última imagen P o I y antes de la imagen B. La palabra de código es la representación binaria natural del número de imágenes no transmitidas más uno. El número máximo de imágenes no transmitidas es 6.

5.1.8 Información de cuantificación para imágenes B (DBQUANT, *quantization information for B-pictures*) (2 bits)

DBQUANT está presente si PTYPE indica «trama PB» (véase también el Anexo G). En el proceso de decodificación, se obtiene un parámetro de cuantificación QUANT para cada macrobloque. Con tramas PB, QUANT se utiliza para el bloque P, mientras que para el bloque B, se utiliza un parámetro de cuantificador diferente, BQUANT. QUANT está comprendido entre 1 y 31. DBQUANT indica la relación entre QUANT y BQUANT, como se define en el Cuadro 3. En este cuadro, «/» significa división por truncamiento. BQUANT está comprendido entre 1 y 31; si el valor de BQUANT resultante del Cuadro 3 es superior a 31, se recorta a 31.

CUADRO 3/H.263

Códigos DBQUANT y relación entre QUANT y BQUANT

DBQUANT	BQUANT
00	$(5 \times \text{QUANT})/4$
01	$(6 \times \text{QUANT})/4$
10	$(7 \times \text{QUANT})/4$
11	$(8 \times \text{QUANT})/4$

5.1.9 Información de inserción suplementaria (PEI, *extra insertion information*) (1 bit)

Un bit que, cuando está puesto a «1», señala la presencia del campo de datos opcional siguiente.

5.1.10 Información de reserva (PSPARE, *spare information*) (0/8/16 ... bits)

Cuando PEI se fija a «1», le siguen 9 bits que consisten en 8 bits de datos (PSPARE) y luego otro bit PEI para indicar si siguen otros 9 bits, y así sucesivamente. Los codificadores no insertarán PSPARE hasta que la UIT lo especifique. Los decodificadores descartarán PSPARE si PEI se fija a 1. Esto permitirá que, en el futuro, la UIT especifique adiciones a PSPARE compatibles hacia atrás. Si a PSPARE le sigue PEI = 0, PSPARE = xx000000 está prohibida, para evitar la emulación del código de comienzo (x = no importa, de manera que 4 de entre 256 valores están prohibidos).

5.1.11 Relleno (ESTUF, *stuffing*) (longitud variable)

Palabra de código de longitud variable que consiste en menos de 8 bits cero. Los codificadores pueden insertar esta palabra de código directamente antes de una palabra de código EOS. Si ESTUF está presente, el último bit de ESTUF será el último bit (menos significativo) de un byte, de modo que el comienzo de la palabra de código EOS está alineada en byte. Los decodificadores serán diseñados de modo que puedan descartar ESTUF.

5.1.12 Fin de secuencia (EOS, *end of sequence*) (22 bits)

Palabra de código de 22 bits. Su valor es 0000 0000 0000 0000 1 11111. Este decodificador decidirá si inserta o no esta palabra de código. EOS puede estar alineada en byte. Esto se puede lograr insertando ESTUF antes del código de comienzo, de manera que el primer bit del código de comienzo sea el primer bit (más significativo) de un byte.

5.1.13 Relleno (PSTUF, *stuffing*) (longitud variable)

Palabra de código de longitud variable que consiste en menos de 8 bits cero. Los codificadores insertarán esta palabra de código para la alineación en byte de la siguiente PSC. El último bit de PSTUF será el último bit (menos significativo) de un byte, de modo que el tren de bits de vídeo que incluye PSTUF es un múltiplo de 8 bits del primer bit en el tren de bits H.263. Los decodificadores se diseñarán de modo que puedan descartar PSTUF.

Si por algún motivo el codificador detiene la codificación de imágenes durante un determinado periodo de tiempo y reanuda la codificación ulteriormente, PSTUF se retransmitirá antes de que el codificador se detenga, para impedir que los últimos 7 bits de la imagen anterior no se envíen hasta que el codificador reanude la codificación.

5.2 Capa de grupo de bloques

Los datos de cada grupo de bloques (GOB) consisten en un encabezamiento GOB, seguido de datos de macrobloque. La estructura se muestra en la Figura 8. Cada GOB contiene una o más filas de macrobloques. En el primer GOB (número 0), no se transmitirá el encabezamiento GOB. En todos los demás GOB, el encabezamiento GOB puede estar vacío, según la estrategia elegida por el codificador. Un decodificador puede señalar al codificador distante por medios externos, por ejemplo, Recomendación H.245, que transmita sólo encabezamientos GOB no vacíos. GSTUF puede estar presente cuando GBSC está presente. GN, GFID y GQUANT están presentes cuando GBSC está presente. GLCI está presente cuando el modo multipunto de presencia continua está activado, como se indica en el encabezamiento de imagen.

GSTUF	GBSC	GN	GSBI	GFID	GQUANT	Datos de macrobloque
-------	------	----	------	------	--------	----------------------

FIGURA 8/H.263

Estructura de la capa GOB

5.2.1 Relleno (GSTUF, *stuffing*) (Longitud variable)

Palabra de código de longitud variable que consiste en menos de 8 bits cero. Los codificadores pueden insertar esta palabra de código directamente antes de una palabra de código GBSC. Si GSTUF está presente, el último bit de GSTUF será el último bit (menos significativo) de un byte, de modo que el comienzo de la palabra de código GBSC esté alineada en byte. Los decodificadores serán diseñados de modo que puedan descartar GSTUF.

5.2.2 Código de comienzo de grupo de bloque (GBSC, *group of block start code*) (17 bits)

Palabra de 17 bits. Su valor es 0000 0000 0000 0000 1. Los códigos de comienzo GOB pueden estar alineados en byte. Esto se consigue insertando GSTUF antes del código de comienzo, de manera que el primer bit del código de comienzo es el primer bit (más significativo) de un byte.

5.2.3 Número de grupo (GN, *group number*) (5 bits)

Palabra de código de longitud fija de 5 bits. Los bits son la representación binaria del número del grupo de bloques. En el caso del GOB numerado 0, el encabezamiento GOB que incluye GSTUF, GBSC, GN, GSBI, GFID y GQUANT está vacío; el número de grupo 0 se utiliza en PSC. El número de grupo 31 se utiliza en EOS, y los valores 18 a 30 están reservados para uso futuro por la UIT.

5.2.4 Indicador de subtren de bits de GOB (GSBI, *GOB sub-bit stream indicator*) (2 bits)

Palabra de código de longitud fija de 2 bits, que sólo está presente si el modo multipunto de presencia continua está indicado por CPM. Las palabras de código son la representación binaria natural del número de canal lógico para el encabezamiento GOB y toda la información siguiente hasta el próximo código de comienzo de imagen o de GOB (véase también el Anexo C).

5.2.5 Identificador de trama GOB (GFID, *GOB frame ID*) (2 bits)

Palabra de código de longitud fija de 2 bits. GFID tendrá el mismo valor en cada encabezamiento GOB de una imagen dada. Además, si el PTYPE indicado en un encabezamiento de imagen es el mismo que el de la imagen transmitida previamente, GFID tendrá el mismo valor que el de esa imagen anterior. Sin embargo, si PTYPE en un determinado encabezamiento de imagen difiere del PTYPE en el encabezamiento de imagen transmitido anteriormente, el valor para GFID en esa imagen diferirá del valor en la imagen previa.

5.2.6 Información de cuantificador (GQUANT, *quantizer information*) (5 bits)

Palabra de código de longitud fija de 5 bits, que indica el cuantificador QUANT que se utilizará para la parte restante de la imagen hasta que sea actualizado por cualquier GQUANT-DQUANT subsiguiente. Las palabras de código son la representación binaria natural de los valores de QUANT que, puesto que tiene un paso mitad, están comprendidos en la gama de 1 a 31.

5.3 Capa de macrobloque

Los datos de cada macrobloque consisten en un encabezamiento de macrobloque, seguido de los datos de bloque. La estructura aparece en la Figura 9. COD sólo está presente en las imágenes para las que PTYPE indica «INTER» para cada macrobloque de esas imágenes. MCBPC está presente cuando lo indica COD o cuando PTYPE indica «INTRA». MODB está presente para el tipo de macrobloque 0-4 si PTYPE indica «trama PB». CBPY, DQUANT, MVD y MVD₂₋₄ están presentes cuando lo indica MCBPC. CBPB y MVDB están presentes sólo si lo indica MODB. «Datos de bloque» está presente cuando lo indican MCBPC y CBPY. MVD₂₋₄ sólo están presentes en el modo predicción avanzada (véase el Anexo F). MODB, CBPB y MVDB sólo están presentes en el modo tramas PB (véase el Anexo G). Para la codificación de los símbolos del modo codificación aritmética basada en sintaxis, véase el Anexo E.

COD	MCBPC	MODB	CBPB	CBPY	DQUANT	MVD	MVD ₂	MVD ₃	MVD ₄	MVDB	Datos de bloque
-----	-------	------	------	------	--------	-----	------------------	------------------	------------------	------	-----------------

FIGURA 9/H.263

Estructura de la capa de macrobloque

5.3.1 Indicación de macrobloque codificado (COD, coded macroblock indication) (1 bit)

Un bit que, cuando está puesto a «0», señala que el bloque está codificado. Cuando está puesto a «1», no se transmite más información para este macrobloque en este caso, el decodificador tratará el macrobloque como un bloque INTER con vector de movimiento para todo el bloque igual a cero y sin datos de coeficiente. COD sólo está presente en las imágenes para las que PTYPE indica «INTER», en cada macrobloque de esas imágenes.

NOTA – En el modo predicción avanzada, se realiza también la compensación de movimiento con superposición si COD está puesto a «1».

5.3.2 Tipo de macrobloque y patrón de grupo codificado para la cromaticidad (MCBPC, macroblock type & coded block pattern for chrominance) (longitud variable)

El patrón de bloque codificado de cromaticidad indica bloques C_B y/o C_R cuando se transmite al menos un coeficiente de transformada no INTRADC (INTRADC es el coeficiente dc para bloques INTRA, véase 5.4.1). $CBPC_N = 1$ si cualquier coeficiente no INTRADC está presente para el bloque N, de lo contrario 0, para $CBPC_5$ y $CBPC_6$ en el patrón de bloque codificado. La numeración de bloques se ofrece en la Figura 5. Cuando MCBPC = relleno, se salta la parte restante de la capa de macrobloque. En este caso, COD = 0 anterior no está relacionado con ningún macrobloque codificado o no codificado y, por consiguiente, no se incrementa el número de macrobloques. En el caso de las imágenes P, los rellenos múltiples se consiguen mediante conjuntos múltiples de COD = 0 y MCBPC = relleno. Véanse los Cuadros 4 y 5.

El patrón de bloque codificado de cromaticidad indica bloques C_B y/o C_R cuando se transmite al menos un coeficiente de transformada no INTRADC. $CBPC_N = 1$ si cualquier coeficiente no INTRADC está presente para el bloque N, de lo contrario 0, para $CBPC_5$ y $CBPC_6$ en el patrón de bloque codificado. La numeración de bloques se ofrece en la Figura 5. Cuando MCBPC = relleno, se salta la parte restante de la capa de macrobloque. En este caso, COD = 0 anterior no está relacionado con ningún macrobloque codificado o no codificado y, por consiguiente, no se incrementa el número de macrobloques. En el caso de las imágenes codificadas en modo INTER, los rellenos múltiples se consiguen mediante conjuntos múltiples de COD = 0 y MCBPC = relleno. Véanse los Cuadros 6 y 7.

CUADRO 4/H.263

Tabla de códigos de longitud variable (VLC) para MCBPC (imágenes I)

Índice	Tipo de macrobloque	CBPC (56)	Número de bits	Código
0	3	00	1	1
1	3	01	3	001
2	3	10	3	010
3	3	11	3	011
4	4	00	4	0001
5	4	01	6	0000 01
6	4	10	6	0000 10
7	4	11	6	0000 11
8	Relleno	–	9	0000 0000 1

5.3.3 Modo macrobloque para bloques B (MODB, macroblock mode for B-blocks) (longitud variable)

MODB está presente para MB-tipo 0-4 si PTYPE indica «trama PB» y es una palabra de código de longitud variable que indica si CBPB está presente (indica que se transmiten coeficientes B para este macrobloque) y/o MVDB está presente. En el Cuadro 8 se definen las palabras de código para MODB.

5.3.4 Patrón de bloque codificado para bloque B (CBPB, coded block pattern for B-blocks) (6 bits)

CBPB sólo está presente en el modo tramas PB si lo indica MODB. $CBPB_N = 1$ si cualquier coeficiente está presente para N bloques B, de lo contrario, 0 para cada bit $CBPB_N$ del patrón de bloque codificado. La numeración de bloques se ofrece en la Figura 5, y el bit del extremo izquierdo de CBPB corresponde al número de bloque 1.

CUADRO 5/H.263

Tabla de códigos de longitud variable (VLC) para MCBPC (imágenes P)

Índice	Tipo de macrobloque	CBPC (56)	Número de bits	Código
0	0	00	1	1
1	0	01	4	0011
2	0	10	4	0010
3	0	11	6	0001 01
4	1	00	3	011
5	1	01	7	0000 111
6	1	10	7	0000 110
7	1	11	9	0000 0010 1
8	2	00	3	010
9	2	01	7	0000 101
10	2	10	7	0000 100
11	2	11	8	0000 0101
12	3	00	5	0001 1
13	3	01	8	0000 0100
14	3	10	8	0000 0011
15	3	11	7	0000 011
16	4	00	6	0001 00
17	4	01	9	0000 0010 0
18	4	10	9	0000 0001 1
19	4	11	9	0000 0001 0
20	Relleno	–	9	0000 0000 1

CUADRO 6/H.263

Tipos de macrobloque y elementos de datos incluidos para las imágenes normales

Tipo de imagen	Tipo de MB	Nombre	COD	MCBPC	CBPY	DQUANT	MVD	MVD ₂₋₄
INTER	No codificado	–	X					
INTER	0	INTER	X	X	X		X	
INTER	1	INTER + Q	X	X	X	X	X	
INTER	2	INTER4V	X	X	X		X	X
INTER	3	INTRA	X	X	X			
INTER	4	INTRA + Q	X	X	X	X		
INTER	Relleno	–	X	X				
INTRA	3	INTRA		X	X			
INTRA	4	INTRA + Q		X	X	X		
INTRA	Relleno	–		X				

NOTA – «X» significa que el ítem está presente en el macrobloque.

CUADRO 7/H.263

Tipos de macrobloque y elementos de datos incluidos para las tramas PB

Tipo de imagen	Tipo de MB	Nombre	COD	MCBPC	MODB	CBPY	CBPB	DQUANT	MVD	MVDB	MVD ₂₋₄
INTER	No co-dificado	-	X								
INTER	0	INTER	X	X	X	X	(X)		X	(X)	
INTER	1	INTER + Q	X	X	X	X	(X)	X	X	(X)	
INTER	2	INTER 4V	X	X	X	X	(X)		X	(X)	X
INTER	3	INTRA	X	X	X	X	(X)		X	(X)	
INTER	4	INTRA + Q	X	X	X	X	(X)	X	X	(X)	
INTER	Relleno	-	X	X							

NOTAS

- «X» significa que el ítem está presente en el macrobloque.
- CBPB y MVDB sólo están presentes si lo indica MODB.
- Los bloques B siempre están codificados en modo INTER, incluso si el tipo de MB del macrobloque PB indica INTRA.

CUADRO 8/H.263

Tabla de códigos de longitud variable (VLC) para MODB

Índice	CBPB	MVDB	Número de bits	Código
0			1	0
1		X	2	10
2	X	X	2	11

NOTA – «X» significa que el ítem está presente en el macrobloque.

5.3.5 Patrón de bloque codificado para la luminancia (CBPY, coded block pattern for luminance) (longitud variable)

Palabra de código de longitud variable, que proporciona un número de patrón que se refiere a aquellos bloques Y del macrobloque para los que se transmite al menos un coeficiente de transformada no INTRADC (INTRADC es el coeficiente dc para bloques INTRA, véase 5.4.1).

$CBPY_N = 1$ si cualquier coeficiente no INTRADC está presente para el bloque N, de lo contrario, 0 para cada bit $CBPY_N$ del patrón de bloque codificado. La numeración de bloques se ofrece en la Figura 5, y el bit del extremo izquierdo de CBPY corresponde al número de bloque 1. Con un patrón $CBPY_N$ determinado se utilizan diferentes palabras de código según si el modo es INTER o INTRA, como se indica en el Cuadro 10.

5.3.6 Información de cuantificador (DQUANT, quantizer information) (2 bits)

Código de 2 bits que define un cambio de QUANT. En el Cuadro 9 se ofrecen los valores diferenciales de las diferentes palabras de código. QUANT tiene una gama de 1 a 31; si después de añadir el valor diferencial, el valor de QUANT es inferior a 1 o superior a 31, se lo recorta a 1 y 31, respectivamente.

CUADRO 9/H.263

Códigos DQUANT y valores diferenciales para QUANT

Índice	Valor diferencial	DQUANT
0	-1	00
1	-2	01
2	1	10
3	2	11

CUADRO 10/H.263

Tabla de códigos de longitud variable (VLC) para CBPY

Índice	CBPY(INTRA) (12 34)	CBPY(INTER) (12 34)	Número de bits	Código
0	00 00	11 11	4	0011
1	00 01	11 10	5	0010 1
2	00 10	11 01	5	0010 0
3	00 11	11 00	4	1001
4	01 00	10 11	5	0001 1
5	01 01	10 10	4	0111
6	01 10	10 01	6	0000 10
7	01 11	10 00	4	1011
8	10 00	01 11	5	0001 0
9	10 01	01 10	6	0000 11
10	10 10	01 01	4	0101
11	10 11	01 00	4	1010
12	11 00	00 11	4	0100
13	11 01	00 10	4	1000
14	11 10	00 01	4	0110
15	11 11	00 00	2	11

5.3.7 Datos de vector de movimiento (MVD, *motion vector data*) (longitud variable)

MVD se incluye en todos los macrobloques INTER (en el modo trama PB, también en los macrobloques INTRA), y consiste en una palabra de código de longitud variable para la componente horizontal, seguida de una palabra de código de longitud variable para la componente vertical. Los códigos de longitud variable se ofrecen en el Cuadro 11.

5.3.8 Datos de vector de movimiento (MVD₂₋₄, *motion vector data*) (longitud variable)

Las tres palabras de código MVD₂₋₄ están incluidas si lo indica PTYPE y MCBPC, y cada una consiste en una palabra de código de longitud variable para el componente horizontal, seguida de una palabra de código de longitud variable para el componente vertical de cada vector. Los códigos de longitud variable se ofrecen en el Cuadro 11. MVD₂₋₄ sólo está presente en el modo predicción avanzada (véase el Anexo F).

5.3.9 Datos de vector de movimiento para macrobloque B (MVDB, *motion vector data for B-macroblock*) (longitud variable)

MVDB sólo está presente en el modo tramas PB si lo indica MODB, y consiste en una palabra de código de longitud variable para el componente horizontal, seguida de una palabra de código de longitud variable para el componente vertical de cada vector. Los códigos de longitud variable se ofrecen en el Cuadro 11. Para el uso de MVDB, véase el Anexo G.

5.4 Capa de bloque

Si no se utiliza el modo tramas PB, un macrobloque comprende cuatro bloques de luminancia y uno de cada uno de los dos bloques de diferencia de color (véase la Figura 5). La estructura de la capa de bloque se muestra en la Figura 10. INTRADC está presente para cada bloque del macrobloque si MCBPC indica MB tipo 3 ó 4 (véanse los Cuadros 4 y 5). TCOEF está presente si es indicado por MCBPC o CBPY.

En el modo tramas PB, un macrobloque comprende doce bloques. Primero los datos para los seis bloques P se transmiten como en el modo H.263 por defecto, después los datos para los seis bloques B. INTRADC está presente para cada bloque P del macrobloque si MCBPC indica MB tipo 3 ó 4 (véanse los Cuadros 4 y 5). INTRADC no está presente para los bloques B. TCOEF está presente para los bloques B si es indicado por MCBPC o CBPY. TCOEF está presente para los bloques B si es indicado por CBPB.

Para la codificación de los símbolos en el modo codificación aritmética basada en sintaxis, véase el Anexo E.

5.4.1 Coeficiente DC de bloque INTRA (INTRADC, *DC coefficient for INTRA blocks*) (8 bits)

Palabra de código de 8 bits. El código 0000 0000 no se utiliza. El código 1000 0000 no se utiliza, y el nivel de reconstrucción de 1024 se codifica como 1111 1111 (véase el Cuadro 12).

5.4.2 Coeficiente de la transformada (TCOEF, *transform coefficient*) (longitud variable)

Los EVENT (evento) que ocurren con más frecuencia se codifican con los códigos de longitud variable que aparecen en el Cuadro 13. El último bit «s» denota el signo de nivel: «0» si es positivo y «1» si es negativo.

Un EVENT es una combinación de la última indicación de coeficiente no cero [LAST (último)]; («0»: no hay más coeficientes no cero en este bloque, «1»: este es el último coeficiente no cero de este bloque), el número de ceros sucesivos que preceden al coeficiente codificado [RUN (pasada)], y el último valor no cero del coeficiente codificado [LEVEL (nivel)].

Las combinaciones restantes de (LAST, RUN y LEVEL) se codifican con una palabra de 22 bits, que consiste en 7 bits ESCAPE (escape), 1 bit LAST, 6 bits RUN y 8 bits LEVEL. El uso de esta palabra de 22 bits para codificar las combinaciones enumeradas en el Cuadro 13 no está prohibido. En la palabra LEVEL de 8 bits, no se utilizan los códigos 0000 0000 y 1000 0000. Los códigos para RUN y LEVEL se ofrecen en el Cuadro 14.

6 Proceso de decodificación

6.1 Compensación de movimiento

En esta subcláusula se describe la compensación de movimiento del modo de predicción H.263 por defecto. Para una descripción de la compensación de movimiento para el modo vector de movimiento sin restricción, véase el Anexo D. Para una descripción de la compensación de movimiento del modo predicción avanzada, véase el Anexo F.

CUADRO 11/H.263

Tabla de códigos de longitud variable (VLC) para MVD

Índice	Vector	Diferencias	Número de bit	Códigos
0	-16	16	13	0000 0000 0010 1
1	-15,5	16,5	13	0000 0000 0011 1
2	-15	17	12	0000 0000 0101
3	-14,5	17,5	12	0000 0000 0111
4	-14	18	12	0000 0000 1001
5	-13,5	18,5	12	0000 0000 1011
6	-13	19	12	0000 0000 1101
7	-12,5	19,5	12	0000 0000 1111
8	-12	20	11	0000 0001 001
9	-11,5	20,5	11	0000 0001 011
10	-11	21	11	0000 0001 101
11	-10,5	21,5	11	0000 0001 111
12	-10	22	11	0000 0010 001
13	-9,5	22,5	11	0000 0010 011
14	-9	23	11	0000 0010 101
15	-8,5	23,5	11	0000 0010 111
16	-8	24	11	0000 0011 001
17	-7,5	24,5	11	0000 0011 011
18	-7	25	11	0000 0011 101
19	-6,5	25,5	11	0000 0011 111
20	-6	26	11	0000 0100 001
21	-5,5	26,5	11	0000 0100 011
22	-5	27	10	0000 0100 11
23	-4,5	27,5	10	0000 0101 01
24	-4	28	10	0000 0101 11
25	-3,5	28,5	8	0000 0111
26	-3	29	8	0000 1001
27	-2,5	29,5	8	0000 1011
28	-2	30	7	0000 111
29	-1,5	30,5	5	0001 1
30	-1	31	4	0011
31	-0,5	31,5	3	011
32	0		1	1
33	0,5	-31,5	3	010
34	1	-31	4	0010
35	1,5	-30,5	5	0001 0
36	2	-30	7	0000 110
37	2,5	-29,5	8	0000 1010
38	3	-29	8	0000 1000
39	3,5	-28,5	8	0000 0110
40	4	-28	10	0000 0101 10
41	4,5	-27,5	10	0000 0101 00
42	5	-27	10	0000 0100 10
43	5,5	-26,5	11	0000 0100 010
44	6	-26	11	0000 0100 000
45	6,5	-25,5	11	0000 0011 110
46	7	-25	11	0000 0011 100
47	7,5	-24,5	11	0000 0011 010
48	8	-24	11	0000 0011 000
49	8,5	-23,5	11	0000 0010 110
50	9	-23	11	0000 0010 100
51	9,5	-22,5	11	0000 0010 010
52	10	-22	11	0000 0010 000
53	10,5	-21,5	11	0000 0001 110
54	11	-21	11	0000 0001 100
55	11,5	-20,5	11	0000 0001 010
56	12	-20	11	0000 0001 000
57	12,5	-19,5	12	0000 0000 1110
58	13	-19	12	0000 0000 1100
59	13,5	-18,5	12	0000 0000 1010
60	14	-18	12	0000 0000 1000
61	14,5	-17,5	12	0000 0000 0110
62	15	-17	12	0000 0000 0100
63	15,5	-16,5	13	0000 0000 0011 0

INTRADC	TCOEF
---------	-------

FIGURA 10/H.263
Estructura de la capa de bloque

CUADRO 12/H.263
Niveles de reconstrucción del coeficiente DC para el modo INTRA

Índice	FLC	Nivel de reconstrucción a transformada inversa
0	0000 0001 (1)	8
1	0000 0010 (2)	16
2	0000 0011 (3)	24
.	.	.
.	.	.
126	0111 1111 (127)	1016
127	1111 1111 (255)	1024
128	1000 0001 (129)	1032
.	.	.
.	.	.
252	1111 1101 (253)	2024
253	1111 1110 (254)	2032
FLC Código de longitud fija.		

6.1.1 Vectores de movimiento diferenciales

El vector de macrobloque se obtiene añadiendo predictores a las diferencias vectoriales indicadas por MVD (véase el Cuadro 11). Para la codificación diferencial de cuatro vectores por macrobloque, véase el Anexo F. En el caso de un vector por macrobloque, los predictores candidatos para la codificación diferencial se toman de los tres macrobloques adyacentes, como se indica en la Figura 11. Los predictores se calculan por separado para las componentes horizontal y vertical.

En los casos especiales que se dan en los bordes de la imagen o el GOB objeto de predicción, rigen las reglas de decisión siguientes, en orden creciente:

- 1) Cuando el macrobloque correspondiente se codificó en modo INTRA (si no está en el modo tramas PB) o no se codificó (COD = 1), el predictor candidato se fija a cero.
- 2) El predictor candidato MV1 se fija a cero si el macrobloque correspondiente está fuera de la imagen (a la izquierda).
- 3) Entonces, los predictores candidatos MV2 y MV3 se fijan a MV1 si los macrobloques correspondientes están fuera de la imagen (parte superior) o fuera de GOB (parte superior) si el encabezamiento GOB del GOB objeto de predicción no está vacío.
- 4) Entonces, el predictor candidato MV3 se fija a cero, si el macrobloque correspondiente está fuera de la imagen (a la derecha).

Para cada componente, el predictor es la mediana de los tres predictores candidatos para esa componente.

Se aprovecha el hecho de que la gama de valores de componentes del vector de movimiento está limitada. Cada palabra VLC representa un par de valores diferenciales. Sólo un elemento del par arrojará un componente de vector de macrobloque que cae dentro de la gama permitida $[-16, 15,5]$. Un valor positivo del componente horizontal o vertical del vector de movimiento significa que la predicción se forma a partir de los pixels de la imagen anterior, que están situados a la derecha o por debajo de los pixels objeto de predicción.

CUADRO 13/H.263

Tabla de códigos de longitud variable (VLC) para TCOEF

ÍNDICE	LAST	RUN	LEVEL	BITS	CÓDIGO VLC
0	0	0	1	3	10s
1	0	0	2	5	1111s
2	0	0	3	7	0101 01s
3	0	0	4	8	0010 111s
4	0	0	5	9	0001 1111s
5	0	0	6	10	0001 0010 1s
6	0	0	7	10	0001 0010 0s
7	0	0	8	11	0000 1000 01s
8	0	0	9	11	0000 1000 00s
9	0	0	10	12	0000 0000 111s
10	0	0	11	12	0000 0000 110s
11	0	0	12	12	0000 0100 000s
12	0	1	1	4	110s
13	0	1	2	7	0101 00s
14	0	1	3	9	0001 1110s
15	0	1	4	11	0000 0011 11s
16	0	1	5	12	0000 0100 001s
17	0	1	6	13	0000 0101 0000s
18	0	2	1	5	1110s
19	0	2	2	9	0001 1101s
20	0	2	3	11	0000 0011 10s
21	0	2	4	13	0000 0101 0001s
22	0	3	1	6	0110 1s
23	0	3	2	10	0001 0001 1s
24	0	3	3	11	0000 0011 01s
25	0	4	1	6	0110 0s
26	0	4	2	10	0001 0001 0s
27	0	4	3	13	0000 0101 0010s
28	0	5	1	6	0101 1s
29	0	5	2	11	0000 0011 00s
30	0	5	3	13	0000 0101 0011s
31	0	6	1	7	0100 11s
32	0	6	2	11	0000 0010 11s
33	0	6	3	13	0000 0101 0100s
34	0	7	1	7	0100 10s
35	0	7	2	11	0000 0010 10s
36	0	8	1	7	0100 01s
37	0	8	2	11	0000 0010 01s
38	0	9	1	7	0100 00s
39	0	9	2	11	0000 0010 00s
40	0	10	1	8	0010 110s
41	0	10	2	13	0000 0101 0101s
42	0	11	1	8	0010 101s
43	0	12	1	8	0010 100s
44	0	13	1	9	0001 1100s
45	0	14	1	9	0001 1011s
46	0	15	1	10	0001 0000 1s
47	0	16	1	10	0001 0000 0s
48	0	17	1	10	0000 1111 1s
49	0	18	1	10	0000 1111 0s
50	0	19	1	10	0000 1110 1s

CUADRO 13/H.263 (fin)

Tabla de códigos de longitud variable (VLC) para TCOEF

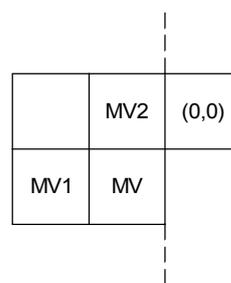
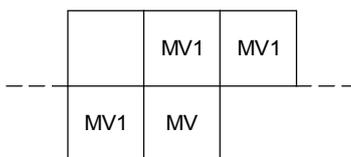
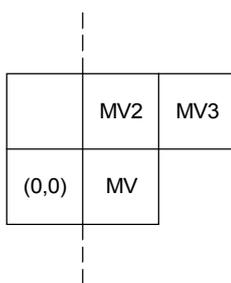
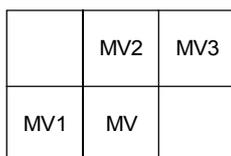
ÍNDICE	LAST	RUN	LEVEL	BITS	CÓDIGO VLC
51	0	20	1	10	0000 1110 0s
52	0	21	1	10	0000 1101 1s
53	0	22	1	10	0000 1101 0s
54	0	23	1	12	0000 0100 010s
55	0	24	1	12	0000 0100 011s
56	0	25	1	13	0000 0101 0110s
57	0	26	1	13	0000 0101 0111s
58	1	0	1	5	0111s
59	1	0	2	10	0000 1100 1s
60	1	0	3	12	0000 0000 101s
61	1	1	1	7	0011 11s
62	1	1	2	12	0000 0000 100s
63	1	2	1	7	0011 10s
64	1	3	1	7	0011 01s
65	1	4	1	7	0011 00s
66	1	5	1	8	0010 011s
67	1	6	1	8	0010 010s
68	1	7	1	8	0010 001s
69	1	8	1	8	0010 000s
70	1	9	1	9	0001 1010s
71	1	10	1	9	0001 1001s
72	1	11	1	9	0001 1000s
73	1	12	1	9	0001 0111s
74	1	13	1	9	0001 0110s
75	1	14	1	9	0001 0101s
76	1	15	1	9	0001 0100s
77	1	16	1	9	0001 0011s
78	1	17	1	10	0000 1100 0s
79	1	18	1	10	0000 1011 1s
80	1	19	1	10	0000 1011 0s
81	1	20	1	10	0000 1010 1s
82	1	21	1	10	0000 1010 0s
83	1	22	1	10	0000 1001 1s
84	1	23	1	10	0000 1001 0s
85	1	24	1	10	0000 1000 1s
86	1	25	1	11	0000 0001 11s
87	1	26	1	11	0000 0001 10s
88	1	27	1	11	0000 0001 01s
89	1	28	1	11	0000 0001 00s
90	1	29	1	12	0000 0100 100s
91	1	30	1	12	0000 0100 101s
92	1	31	1	12	0000 0100 110s
93	1	32	1	12	0000 0100 111s
94	1	33	1	13	0000 0101 1000s
95	1	34	1	13	0000 0101 1001s
96	1	35	1	13	0000 0101 1010s
97	1	36	1	13	0000 0101 1011s
98	1	37	1	13	0000 0101 1100s
99	1	38	1	13	0000 0101 1101s
100	1	39	1	13	0000 0101 1110s
101	1	40	1	13	0000 0101 1111s
102	ESCAPE			7	0000 011

Tabla de códigos de longitud fija (FLC) para RUN Y LEVEL

Índice	Run	Código
0	0	000 000
1	1	000 001
2	2	000 010
.	.	.
.	.	.
63	63	111 111

Índice	Level	Código
-	-128	PROHIBIDO
0	-127	1000 0001
.	.	.
125	-2	1111 1110
126	-1	1111 1111
-	0	PROHIBIDO
127	1	0000 0001
128	2	0000 0010
.	.	.
253	127	0111 1111

El vector de movimiento se usa para todos los pixels de todos los cuatro bloques de luminancia del macrobloque. Los vectores de movimiento de ambos bloques de crominancia se obtienen dividiendo los valores componentes del vector de macrobloque por dos, debido al formato más bajo de la crominancia. Los valores componentes de los vectores resultantes con resolución de un cuarto de pixel se modifican hacia la posición más cercana de mitad de pixel, como se indica en el Cuadro 15.



--- Borde de la imagen o del GOB

- MV Vector de movimiento elegido (*current motion vector*)
- MV1 Vector de movimiento anterior (*previous motion vector*)
- MV2 Vector de movimiento parte superior (*above motion vector*)
- MV3 Vector de movimiento parte superior a la derecha (*above right motion vector*)

T1519590-95/d05

FIGURA 11/H.263
Predicción del vector de movimiento

6.1.2 Interpolación para la predicción de subpixel

Los valores de mitad de pixel se obtienen mediante interpolación bilineal, como se describe en la Figura 12. «/» indica división por truncamiento.

Modificación de los componentes de vectores de crominancia con resolución de un cuarto de pixel

Posición de un cuarto de pixel	0	1/4	1/2	3/4	1
Posición resultante	0	1/2	1/2	1/2	1

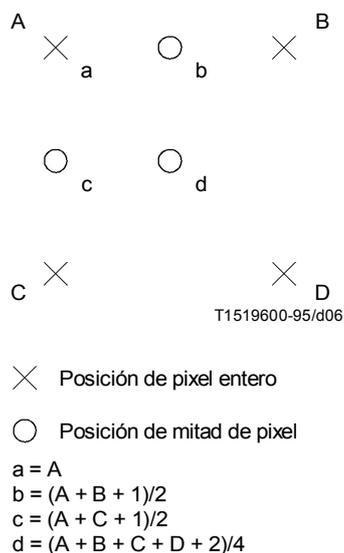


FIGURA 12/H.263

Predicción de mitad de pixel por interpolación bilineal

6.2 Coeficientes de decodificación

6.2.1 Cuantificación inversa

Si LEVEL = «0», entonces el nivel de reconstrucción es REC = «0». El nivel de reconstrucción de INTRADC viene dado por el Cuadro 12. Los niveles de reconstrucción de todos los coeficientes no cero diferentes de INTRADC vienen dados por las fórmulas siguientes:

$$|REC| = QUANT \cdot (2 \cdot |LEVEL| + 1) \quad \text{si QUANT} = \text{«impar»}$$

$$|REC| = QUANT \cdot (2 \cdot |LEVEL| + 1) - 1 \quad \text{si QUANT} = \text{«par»}$$

Nótese que este proceso no autoriza ni siquiera los números con valor. Se ha observado que esto previene la acumulación de errores por desajuste de la transformada discreta de coseno inversa (IDCT, *inverse discrete cosine transform*). Después de calcular |REC|, se añade el signo para obtener REC: $REC = \text{sign}(LEVEL) \cdot |REC|$

Sign (LEVEL) viene dado por el último bit del código TCOEF (véase el Cuadro 13) o por el Cuadro 14.

6.2.2 Recorte de los niveles de reconstrucción

Después de la cuantificación inversa, se recortan los niveles de reconstrucción de todos los coeficientes diferentes de INTRADC para que caigan en la gama -2048 a 2047.

6.2.3 Colocación en zig-zag

Los coeficientes de la transformada cuantificados se colocan en un bloque de 8×8 conforme a la secuencia que aparece en la Figura 13. El coeficiente 1 es el coeficiente dc.

1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

FIGURA 13/H.263

Colocación en zigzag de los coeficientes de la transformada cuantificados

6.2.4 Transformada inversa

Después de la cuantificación inversa y la colocación en zigzag de los coeficientes, los bloques de 8×8 resultantes se procesan con una transformada discreta de coseno bidimensional separable de dimensión 8×8 . La salida de la transformada inversa varía entre -256 y $+255$ después del recorte para poder ser representada mediante 9 bits. La función de transferencia de la transformada inversa viene dada por:

$$f(x, y) = 1/4 \sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v)F(u, v)\cos [\pi(2x + 1)u/16] \cos [\pi(2y + 1)v/16]$$

siendo $u, v, x, y = 0, 1, 2 \dots, 7$

donde:

x, y son las coordenadas espaciales en el dominio de pixel;

u, v son las coordenadas en el dominio de la transformada;

$C(u) = 1/\sqrt{2}$ para $u = 0, 1$ en los demás casos;

$C(v) = 1/\sqrt{2}$ para $v = 0, 1$ en los demás casos.

NOTA – En los bloques sometidos a transformación, $x = 0$ e $y = 0$ se refieren a los pixels más próximos a los bordes izquierdo y superior de la imagen, respectivamente.

Los procedimientos aritméticos para calcular la transformada inversa no se definen, pero ésta debe mantenerse dentro de las tolerancias de error especificadas en el Anexo A.

6.3 Reconstrucción de los bloques

6.3.1 Resumen

Después de la compensación de movimiento y la decodificación de los coeficientes (incluida la transformada inversa), se hace una reconstrucción de cada bloque de luminancia y crominancia. En el caso de los bloques INTRA, la reconstrucción es igual al resultado de la transformación inversa. En el caso de los bloques INTER, la reconstrucción se efectúa sumando la predicción y el resultado de la transformación inversa. La suma se efectúa pixel por pixel.

6.3.2 Recorte

Para evitar que la distorsión de cuantificación de las amplitudes de los coeficientes de transformada causen desbordamiento aritmético en los bucles del codificador y decodificador, se insertan funciones de recorte. El recortador actúa después de la suma de la predicción, y si el error de predicción reconstruido de los valores de pixel resultantes es inferior a 0 o superior a 255, los cambia a 0 y 255, respectivamente.

Anexo A

Especificación de la exactitud de la transformada inversa

(Este anexo es parte integrante de esta Recomendación)

A.1 Se generan valores de datos de pixel enteros aleatorios en la gama de $-L$ a $+H$ de acuerdo con el generador de números aleatorios que se ofrece a continuación (versión «C»). Se disponen en bloques de 8×8 . Deben generarse conjuntos de datos de 10 000 bloques cada uno para ($L = 256, H = 255$), ($L = H = 5$) y ($L = H = 300$).

A.2 Para cada bloque de 8×8 se calcula la transformada discreta directa de coseno separable, ortonormal y de multiplicación matricial con una exactitud de coma flotante de 64 kbit/s por lo menos:

$$F(u, v) = 1/4 C(u) C(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos [\pi(2x + 1)u/16] \cos [\pi(2y + 1)v/16]$$

con $u, v, x, y = 0, 1, 2, \dots, 7$

donde:

x, y son las coordenadas espaciales en el dominio de pixel;

u, v son las coordenadas en el dominio de la transformada;

$C(u) = 1/\sqrt{2}$ para $u = 0, 1$ en los demás casos;

$C(v) = 1/\sqrt{2}$ para $v = 0, 1$ en los demás casos.

A.3 Para cada bloque se redondean a los valores enteros más cercanos los 64 coeficientes resultantes de la transformada. Se recortan a la gama de -2048 a $+2047$, lo que constituye los datos de entrada de 12 bits para la transformada inversa.

A.4 Para cada bloque de 8×8 de datos de 12 bits producido según A.3, se calcula una transformada discreta de coseno inversa (IDCT, *inverse discrete transform*) separable, ortonormal y de multiplicación matricial con una exactitud de coma flotante de 64 bits por lo menos. Se redondean los pixels resultantes al valor entero más cercano y se recortan para que caigan en la gama de -256 a $+255$. Estos bloques de 8×8 pixels constituyen los datos de salida de la IDCT de referencia.

A.5 Para cada bloque de 8×8 de datos producido según A.3, se aplica la IDCT objeto de prueba y se recorta la salida para que caiga en la gama de -256 a $+255$. Estos bloques de 8×8 pixels constituyen los datos de salida de la IDCT de prueba.

A.6 Para cada uno de los 64 pixels de salida de la IDCT y cada uno de los conjuntos de datos de 10 000 bloques generados anteriormente, se miden el error máximo, el error medio y el error cuadrático medio entre los datos de referencia y los de prueba.

- A.7**
- Para cualquier pixel, el valor del error máximo no rebasará la unidad.
 - Para cualquier pixel, el valor del error cuadrático medio no rebasará 0,06.
 - En total, el valor del error cuadrático medio no rebasará 0,02.
 - Para cualquier pixel, el valor del error medio no rebasará 0,015.
 - En total, el valor del error medio no rebasará 0,0015.

A.8 Una entrada «todos ceros» debe producir una salida «todos ceros».

A.9 Se vuelven a realizar las medidas utilizando exactamente los mismos valores de datos de A.1, pero cambiando el signo de cada pixel.

Programa «C» para generación de números aleatorios

/ L y H son números enteros largos, es decir de 32 bits */*

```
long rand (L,H)
long      L,H;
{
```

```

static long   randx = 1;           /* largo es 32 bits      */
static double z = (double) 0x7ffffff;

long         i,j;
double       x;                   /* doble es 64 bits      */

randx = (randx * 1103515245) + 12345;
i = randx & 0x7ffffffe;          /* conservar 30 bits     */
x = ( (double)i ) / z;           /* gama de 0 a 0.99999 ... */
x *= (L+H+1);                   /* gama de 0 a < L+H+1   */
j = x;                          /* conservar la parte entera */
return(j - L);                  /* gama de -L a H       */
}

```

Anexo B

Decodificador ficticio de referencia

(Este anexo es parte integrante de esta Recomendación)

El decodificador ficticio de referencia (HRD, *hypothetical reference decoder*) se define como sigue.

B.1 El HRD y el codificador tienen la misma frecuencia de reloj y la misma velocidad CIF, y funcionan síncronamente.

B.2 El tamaño de la memoria tampón receptora de HRD es $(B + \text{BPPmaxKb} * 1024 \text{ bits})$ donde $(\text{BPPmaxKb} * 1024)$ es el número máximo de bits por imagen que ha sido negociado para uso en el tren de bits (véase 3.6). El valor de B se define como sigue:

$$B = 4 \cdot R_{\text{máx}} / 29,97$$

donde 29,97 es en Hz y $R_{\text{máx}}$ es la velocidad binaria de vídeo máxima durante la conexión en bits por segundo. Este valor para B es un mínimo. Un codificador puede utilizar un valor mayor para B a condición de que el número más grande sea negociado primero por medios externos, por ejemplo, aplicando la Recomendación H.245.

El valor para $R_{\text{máx}}$ depende de la configuración del sistema (por ejemplo red telefónica general conmutada o RDSI, monoenlace o multienlace) y puede ser igual a la velocidad binaria máxima admitida por el enlace físico. La negociación de $R_{\text{máx}}$ se efectúa por medios externos, por ejemplo, aplicando la Recomendación H.245.

B.3 El HRD está vacío inicialmente.

B.4 La memoria tampón del HRD se examina a intervalos de CIF (1000/29,97 ms). Si por lo menos una imagen codificada completa está en la memoria tampón, todos los datos para la imagen anterior se suprimen instantáneamente (por ejemplo, a t_{n+1} en la Figura B.1). Inmediatamente después de suprimir los datos anteriores, la ocupación de la memoria tampón debe ser menor que B . Esto es un requisito del tren de bits de salida del codificador que incluye datos de imagen codificados y MCBPC y relleno STUF pero no bits de alineación de trama para corrección de errores, indicador de relleno (Fi) bits de relleno o información de paridad para corrección de errores, descritos en el Anexo H.

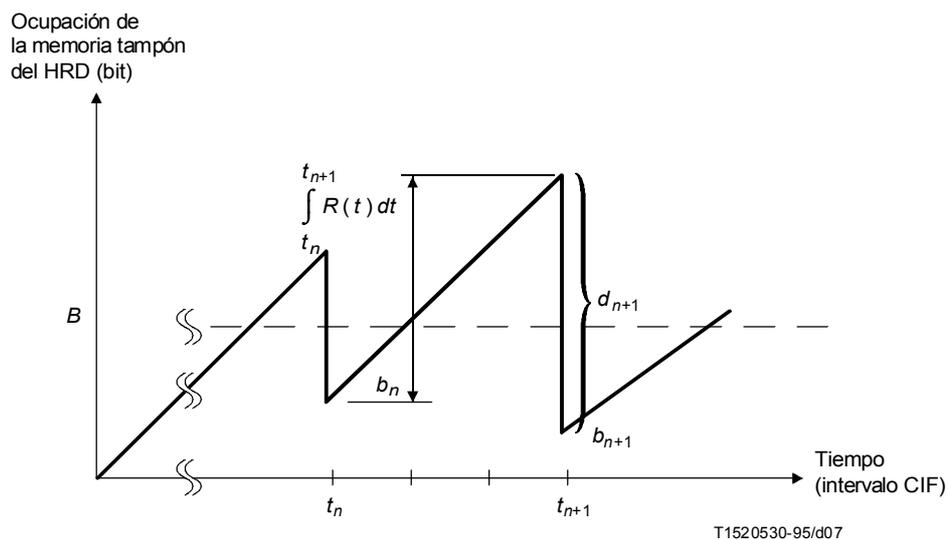
A los efectos de esta definición, una imagen codificada completa es una imagen I o P normal o una trama PB.

Para cumplir este requisito, el número de bits para la $(n+1)$ ésima imagen codificada de d_{n+1} , debe satisfacer:

$$d_{n+1} \geq b_n + \int_{t_n}^{t_{n+1}} R(t)dt - B$$

donde:

- b_n es la ocupación de la memoria tampón justamente después del instante t_n ;
- t_n es el instante en el que la n ésima imagen codificada es suprimida de la memoria tampón del HRD;
- $R(t)$ es la velocidad binaria de vídeo en el instante t .



NOTA – El instante $(t_{n+1} - t_n)$ es un número entero de periodos de imagen CIF (1/29,97, 2/29,97, 3/29,97, ...).

FIGURA B.1/H.263
Ocupación de la memoria tampón del HRD

Anexo C

Consideraciones sobre el funcionamiento multipunto

(Este anexo es parte integrante de esta Recomendación)

Las facilidades que se indican a continuación se proporcionan para soportar el funcionamiento multipunto con conmutación.

C.1 Petición de imagen congelada

Hace que el decodificador congele la imagen visualizada hasta que se reciba una señal de liberación de imagen congelada o hasta que expire un periodo de temporización de al menos seis segundos. Esta señal se transmite por medios externos (por ejemplo, Recomendación H.245).

C.2 Petición de actualización rápida

Hace que el codificador codifique la imagen siguiente en modo INTRA con los parámetros de codificación adecuados para evitar el desbordamiento de la memoria tampón. Esta señal se transmite por medios externos (por ejemplo, Recomendación H.245).

C.3 Liberación de imagen congelada

Señal procedente de un codificador que ha respondido a una petición de actualización rápida y que permite al decodificador salir del modo de imagen congelada y presentar las imágenes decodificadas de manera normal. Esta señal es transmitida por PTYP (véase 5.1.3) en el encabezamiento de imagen de la primera imagen codificada en respuesta a la petición de actualización rápida.

C.4 Multipunto de presencia continua (CPM, *continuous presence multipoint*) (no se utiliza para la Recomendación H.324)

En la presente Recomendación se proporciona un modo multipunto de presencia continua en el cual se puede multiplexar hasta cuatro trenes de bits QCIF H.263 independientes, como «subtrenes» independientes en un nuevo tren de vídeo con los campos PSBI y GSBI. El intercambio de capacidades para este modo se efectúa por medios externos (por ejemplo, la Recomendación H.242).

Cuando se funciona en el modo CPM el campo CPM se pondrá a «1» en cada uno de los trenes de bits H.263 independientes. Los indicadores de subtren de bits (SBI, *sub-bit stream indicators*) sub-bit stream indicators) en la imagen y los encabezamientos GOB de cada tren de bits H.263 indican el número del tren de bits H.263 al cual pertenece ese encabezamiento y toda la información que sigue hasta la próxima imagen o encabezamiento GOB en el tren de bits de vídeo compuesto.

Cada subtren de bits se considera como un tren de bits H.263 normal y por consiguiente satisfará las capacidades intercambiadas por medios externos. La información para los diferentes trenes de bits H.263 no se transmite en ningún orden predefinido especial, un SIB puede tener cualquier valor independientemente de los SIB precedentes y las velocidades de imagen para los diferentes trenes de bits H.263 pueden ser diferentes. La información en cada tren de bits es también completamente independiente con respecto a la información en los otros trenes de bits. Por ejemplo, las palabras de código GFID en un subtren de bits no están influidas por las palabras de código GFID o PTYPE en otros subtrenes de bits.

Anexo D

Modo vector de movimiento sin restricción

(Este anexo es parte integrante de esta Recomendación)

Este anexo describe el modo vector de movimiento sin restricción facultativo de la Recomendación H.263. La capacidad de este modo de la Recomendación H.263 se señala por medios externos (por ejemplo, aplicando la Recomendación H.245). La utilización de este modo se indica en PTYPE.

D.1 Vectores de movimiento en fronteras de imagen

En el modo predicción por defecto de H.263, los vectores de movimiento están limitados, de manera que todos los pixels referenciados por ellos se encuentran dentro del área de la imagen codificada (véase 4.2.3). Ahora bien, en el modo vector de movimiento sin restricción se suprime esta restricción y, por tanto, los vectores de movimiento *están autorizados* a apuntar fuera de la imagen. Cuando un pixel referenciado mediante un vector de movimiento está fuera del área de la imagen codificada, se utiliza en su lugar un pixel de borde. Este pixel de borde se encuentra limitando el vector de movimiento a la última posición de pixel entero dentro del área de la imagen codificada. La limitación del vector de movimiento se realiza de pixel en pixel y por separado para cada componente del vector de movimiento.

Por ejemplo, si se utiliza el modo vector de movimiento sin restricción para una imagen QCIF, el valor de pixel referenciado para la componente de luminancia viene dado por la siguiente fórmula:

$$\text{Rumv}(x, y) = \text{R}(x', y')$$

donde:

x, y, x', y' = coordenadas espaciales en el dominio de pixel;

$\text{Rumv}(x, y)$ = valor de pixel de la imagen referenciada en (x, y) en el modo vector de movimiento sin restricción;

$\text{R}(x', y')$ = valor de pixel de la imagen referenciada en (x, y) en el modo vector de movimiento sin restricción;

$x' = 0$ si $x < 0$;

$x' = 175$ si $x > 175$;

$x' = x$ en los demás casos;

$$\begin{aligned}
 y' &= 0 && \text{si } y < 0; \\
 y' &= 143 && \text{si } y > 143; \\
 y' &= y && \text{en los demás casos;}
 \end{aligned}$$

y la zona de imagen codificada de $R(x', y')$ es $0 \leq x' \leq 175,0 \leq y' \leq 143$. Las fronteras dadas son posiciones de pixel de enteros; sin embargo, (x', y') pueden ser también una mitad de posición de pixel dentro de estas fronteras.

D.2 Extensión de la gama de vectores de movimiento

En el modo de predicción por defecto, los valores para componentes horizontales y verticales de los vectores de movimiento están restringidos a la gama $[-16, 15,5]$ (esto es válido también para los componentes de vectores de movimiento hacia adelante y hacia atrás para imágenes B). Sin embargo, en el modo vector de movimiento sin restricción, la gama máxima para componentes de vectores es $[-31,5, 31,5]$ con la restricción de que sólo se pueden alcanzar los valores que están dentro de una gama de $[-16, 15,5]$ alrededor del predictor para cada componente de vector de movimiento si el predictor está en la gama $[-15,5, 16]$. Si el predictor está fuera de la gama $[-15,5, 16]$ se pueden alcanzar todos los valores dentro de la gama $[-31,5, 31,5]$ con el mismo signo que el predictor más el valor cero. De este modo, si MV_c es el componente de vector de movimiento y P_c es el predictor para el mismo,

$$\begin{aligned}
 -31,5 \leq MV_c \leq 0 & && \text{si } -31,5 \leq P_c \leq -16 \\
 -16 + P_c \leq MV_c \leq 15,5 + P_c & && \text{si } -15,5 \leq P_c \leq 16 \\
 0 \leq MV_c \leq 31,5 & && \text{si } 16,5 \leq P_c \leq 31,5
 \end{aligned}$$

En el modo de vector de movimiento sin restricción, la interpretación del Cuadro 10 para MVD, MVD₂₋₄ y MVDB es como sigue:

- Si el predictor para el componente de vector de movimiento está en la gama $[-15,5, 16]$ solamente se aplica la primera columna de las diferencias de vector.
- Si el predictor para el componente de vector de movimiento está fuera de la gama $[-15,5, 16]$, se utilizará la diferencia de vector del Cuadro 10, lo que resulta en un componente de vector dentro de la gama $[-31,5, 31,5]$ con el mismo signo que el predictor (incluido cero).

El predictor para MV_D y MVD₂₋₄ se define como el valor mediano de los componentes de vector MV1, MV2 y MV3 definidos en 6.1.1 y en F.2. Para MVDB, el predictor $P_c = (TR_B \times MV) / TR_D$ donde MV representa un componente de vector para un bloque de luminancia $8 * 8$ en una imagen P (véase también G.4).

Anexo E

Modo codificación aritmética basada en sintaxis

(Este anexo es parte integrante de esta Recomendación)

E.1 Introducción

En la codificación/decodificación de longitud variable (VLC/VLD, *variable length coding/decoding*), como se describe en la cláusula 5, un símbolo se codifica con VLC mediante un cuadro específico basado en la sintaxis del codificador. Por lo común, ese cuadro contiene las longitudes y los valores de las palabras de código VLC. El símbolo se hace corresponder con una entrada del cuadro en una operación de consulta de cuadro, y luego la palabra de código binario especificada por la entrada se envía normalmente a una memoria tampón para la transmisión al receptor. En la decodificación VLD, el tren binario recibido se empareja, entrada por entrada, con un cuadro específico basado en la sintaxis del codificador. Este cuadro tiene que ser el mismo que el usado en el codificador para codificar el símbolo objeto de decodificación. Después la entrada emparejada en el cuadro se hace corresponder inversamente con el símbolo correspondiente, lo que constituye el resultado final del decodificador VLD y se utiliza más adelante para recuperar las imágenes de vídeo. Este proceso VLC/VLD implica que cada símbolo tiene que ser codificado con un número total fijo de bits. La supresión de esta restricción de número total fijo de bits para los símbolos permite reducir la velocidad binaria resultante, lo que se consigue mediante codificación aritmética.

En este anexo se describe el modo codificación aritmética basada en sintaxis (SAC, *syntax-based arithmetic coding*) opcional de H.263. En este modo, todas las operaciones correspondientes H.263 de codificación/decodificación de longitud variable se sustituyen por operaciones de codificación/decodificación aritmética. La capacidad de este modo H.263 se señala con medios externos (por ejemplo, Recomendación H.245). La utilización de este modo se indica en PTYPE.

E.2 Especificación del codificador SAC

En el modo SAC, un símbolo se codifica utilizando un conjunto específico de números enteros (o un modelo) basado en la sintaxis del codificador y llamando al procedimiento que se ofrece a continuación, especificado en el lenguaje C.

```
#define    q1    16384
#define    q2    32768
#define    q3    49152
#define    top   65535

static long    low, high, opposite_bits, length;

void    encode_a_symbol(int index, int cumul_freq[ ])
{
    length = high - low + 1;
    high = low - 1 + (length * cumul_freq[index]) / cumul_freq[0];
    low += (length * cumul_freq[index+1]) / cumul_freq[0];
    for ( ; ; ) {
        if (high < q2) {
            send out a bit "0" to PSC_FIFO;
            while (opposite_bits > 0) {
                send out a bit "1" to PSC_FIFO;
                opposite_bits--;
            }
        }
        else if (low >= q2) {
            send out a bit "1" to PSC_FIFO;
            while (opposite_bits > 0) {
                send out a bit "0" to PSC_FIFO;
                opposite_bits--;
            }
            low -= q2;
            high -= q2;
        }
        else if (low >= q1 && high < q3) {
            opposite_bits += 1;
            low -= q1;
            high -= q1;
        }
        else break;

        low *= 2;
        high = 2 * high + 1;
    }
}
```

Los valores low, high y opposite_bits se inicializan con 0, top y 0, respectivamente. PSC_FIFO es una memoria FIFO [primero en entrar, primero en salir (*first in first out*)] que almacena los bits de salida del codificador aritmético. El modelo se especifica mediante cumul_freq[], y el símbolo se especifica con el índice del modelo.

E.3 Especificación del decodificador SAC

En el decodificador SAC, un símbolo se decodifica utilizando un modelo específico basado en la sintaxis y llamando al procedimiento que se ofrece a continuación, especificado en el lenguaje C.

```
static long    low, high, code_value, bit, length, index, cum;

int    decode_a_symbol(int cumul_freq[ ])
{
    length = high - low + 1;
    cum = (-1 + (code_value - low + 1) * cumul_freq[0]) / length;
    for (index = 1; cumul_freq[index] > cum; index++);
}
```

```

high = low - 1 + (length * cumul_freq[index-1]) / cumul_freq[0];
low += (length * cumul_freq[index]) / cumul_freq[0];
for ( ; ) {
    if (high < q2);
    else if (low >= q2) {
        code_value -= q2;
        low -= q2;
        high -= q2;
    }
    else if (low >= q1 && high < q3) {
        code_value -= q1;
        low -= q1;
        high -= q1;
    }
    else break;

    low *= 2;
    high = 2 * high + 1;
    get bit from PSC_FIFO;
    code_value = 2 * code_value + bit;
}
return (index-1);
}
}

```

Una vez más, el modelo se especifica mediante `cumul_freq[]`. El símbolo decodificado se devuelve a través de su índice en el modelo. `PSC_FIFO` es una memoria FIFO que almacena el tren de datos entrante. El decodificador se inicializa para empezar la decodificación de un tren binario de codificación aritmética llamando al procedimiento que se ofrece a continuación.

```

void decoder_reset()
{
    code_value = 0;
    low = 0;
    high = top;
    for (int i = 1; i <= 16; i++) {
        get bit from PSC_FIFO;
        code_value = 2 * code_value + bit;
    }
}

```

E.4 Sintaxis

Como en el modo tabla VLC de H.263, la sintaxis de los símbolos se distribuye en cuatro capas: imagen, grupo de bloques, macrobloque y bloque. La sintaxis de las tres capas superiores se mantiene exactamente igual. La sintaxis de la capa de bloque también es bastante similar, pero se muestra en la Figura E.1.

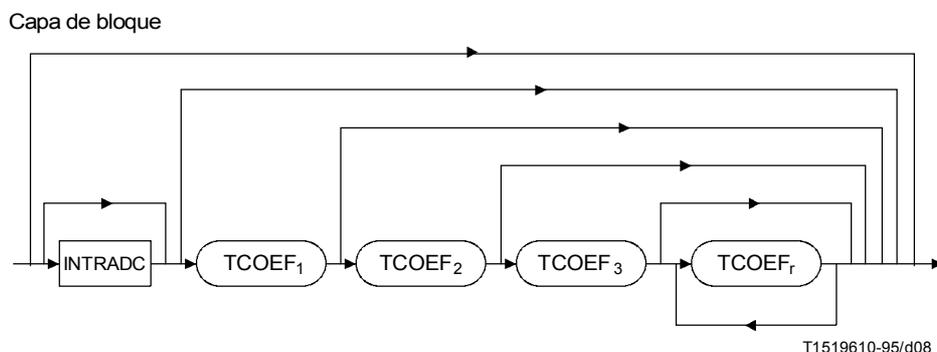


FIGURA E.1/H.263
Estructura de la capa de bloque SAC

En la Figura E.1, TCOEF₁, TCOEF₂, TCOEF₃ y TCOEF_r son símbolos de LAST-RUN-LEVEL, como se define en 5.4.2, y son, respectivamente, los posibles 1º, 2º, 3º y resto de los símbolos. TCOEF₁, TCOEF₂, TCOEF₃ y TCOEF_r sólo están presentes si uno, dos, tres o más coeficientes están presentes en la capa de bloque, respectivamente.

E.5 PSC_FIFO

PSC_FIFO en el codificador o en el decodificador es una FIFO de tamaño superior a 17 bits. En la PSC_FIFO del codificador, se localizan las emulaciones ilegales de PSC y GBSC y se evitan con un relleno de «1» después de cada aparición sucesiva de 14 «0» (que no forman parte de PSC o de GBSC). En la PSC_FIFO del decodificador, se suprime el primer «1» después de cada serie de 14 «0»; cuando una serie de 14 «0» está seguida por un «0», esto indica que se ha detectado un PSC o GBSC legal. La localización exacta de PSC o GBSC viene determinada por el próximo «1» que sigue a la cadena de ceros.

E.6 Símbolos de longitud fija

Los símbolos de longitud fija forman tres cadenas posibles, PSC--TR--PTYPE--PQUANT--CPM--(PSBI)--(TRB-DBQUANT)--PEI--(PSPARE--PEI--...), (GSTUF)--GBSC--GN--(GSBI)--GFID--GQUANT y (ESTUF)--(EOS)--(PSTUF). Estas cadenas se envían directamente a PSC_FIFO como en el modo tabla VLC normal de H.263 en el lado del codificador, y se envían directamente desde la PSC_FIFO del decodificador tras la detección de PSC o GBSC o EOS legales.

Si una cadena de símbolos de longitud fija no es la primera en una sesión de vídeo, hay que reinicializar el codificador aritmético antes de enviar la cadena de símbolos de longitud fija llamando al procedimiento que se ofrece a continuación. Este procedimiento también se llamará al final de una sesión de vídeo o antes de EOS.

```
void encoder_flush()
{
    opposite_bits++;
    if (low < q1) {
        send out a bit "0" to PSC_FIFO;
        while (opposite_bits > 0) {
            send out a bit "1" to PSC_FIFO;
            opposite_bits--;
        }
    }
    else {
        send out a bit "1" to PSC_FIFO;
        while (opposite_bits > 0) {
            send out a bit "0" to PSC_FIFO;
            opposite_bits--;
        }
    }
    low = 0;
    high = top;
}
```

En el decodificador, después de cada cadena de símbolos de longitud fija, se llama al procedimiento **decoder_reset**.

E.7 Símbolo de longitud no fija

En E.8 se incluyen modelos para los símbolos de longitud no fija. Los índices que se indican en las tablas VLC de la cláusula 5 se utilizan para indizar los números enteros de los modelos.

Los modelos para COD y MCBPC en imágenes P son nombrados por cumf_COD y cumf_MCBPC. El índice de COD es «0» cuando está en cero, y «1» cuando está en 1. Los índices de MCBPC se definen en el Cuadro 4 para tramas INTRA y en el Cuadro 5 para imágenes P. El modelo de MCBPC de imágenes I es nombrado por cumf_MCBPC_intra.

El modelo de MODB es cumf_MODB. Los índices de MODB se definen en el Cuadro 8. El modelo de CBPB_n, con n=1,2,...,4, es cumf_YCBPB, y el modelo de CBPB_n, con n=5,6, es cumf_UVCBPB, y los índices son 0 para CBPB_n=0 y 1 para CBPB_n=1.

El modelo de CBPY es cumf_CBPY para macrobloque INTER y cumf_CBPY_intra para macrobloque INTRA. El modelo de DQUANT es cumf_DQUANT. La indización para CBPY y DQUANT se define en los Cuadros 10 y 9, respectivamente.

El modelo de MVD, MVD_{2,4} y MVDB es cumf_MVD y el modelo de INTRADC es cumf_INTRADC. La indización se define en los Cuadros 11 y 12, respectivamente.

Un TCOEF sin escape consiste en un símbolo para TCOEF1/2/3/r, seguido de un símbolo, SIGN, para el signo del TCOEF. Los modelos de TCOEF1, TCOEF2, TCOEF3 y TCOEFr para bloque INTER son cumf_TCOEF1, cumf_TCOEF2, cumf_TCOEF3, cumf_TCOEFr. Los modelos de bloque INTRA son cumf_TCOEF1_intra, cumf_TCOEF2_intra, cumf_TCOEF3_intra, cumf_TCOEFr-intra. Para todos los TCOEF, la indización se define en el Cuadro 13. El modelo de SIGN es cumf_SIGN. La indización de SIGN es 0 para signo positivo y 1 para signo negativo.

Los modelos de LAST, RUN, LEVEL después de ESCAPE son cumf_LAST (cumf_LAST_intra), cumf_RUN (cumf_RUN_intra), cumf_LEVEL (cumf_LEVEL_intra) para bloque INTER (INTRA). La indización de LAST es 0 para LAST=0 y 1 LAST=1, mientras que la indización de RUN y LEVEL se define en el Cuadro 14.

E.8 Modelos SAC

int cumf_COD[3]={16383, 6849, 0};

int cumf_MCBPC[22]={16383, 4105, 3088, 2367, 1988, 1621, 1612, 1609, 1608, 496, 353, 195, 77, 22, 17, 12, 5, 4, 3, 2, 1, 0};

int cumf_MCBPC_intra[10]={16383, 7410, 6549, 5188, 442, 182, 181, 141, 1, 0};

int cumf_MODB[4]={16383, 6062, 2130, 0};

int cumf_YCBPB[3]={16383, 6062, 0};

int cumf_UVCBPB[3]={16383, 491, 0};

int cumf_CBPY[17]={16383, 14481, 13869, 13196, 12568, 11931, 11185, 10814, 9796, 9150, 8781, 7933, 6860, 6116, 4873, 3538, 0};

int cumf_CBPY_intra[17]={16383, 13619, 13211, 12933, 12562, 12395, 11913, 11783, 11004, 10782, 10689, 9928, 9353, 8945, 8407, 7795, 0};

int cumf_DQUANT[5]={16383, 12287, 8192, 4095, 0};

int cumf_MVD[65]={16383, 16380, 16369, 16365, 16361, 16357, 16350, 16343, 16339, 16333, 16326, 16318, 16311, 16306, 16298, 16291, 16283, 16272, 16261, 16249, 16235, 16222, 16207, 16175, 16141, 16094, 16044, 15936, 15764, 15463, 14956, 13924, 11491, 4621, 2264, 1315, 854, 583, 420, 326, 273, 229, 196, 166, 148, 137, 123, 114, 101, 91, 82, 76, 66, 59, 53, 46, 36, 30, 26, 24, 18, 14, 10, 5, 0};

int cumf_INTRADC[255]={16383, 16380, 16379, 16378, 16377, 16376, 16370, 16361, 16360, 16359, 16358, 16357, 16356, 16355, 16343, 16238, 16237, 16236, 16230, 16221, 16220, 16205, 16190, 16169, 16151, 16130, 16109, 16094, 16070, 16037, 16007, 15962, 15938, 15899, 15854, 15815, 15788, 15743, 15689, 15656, 15617, 15560, 15473, 15404, 15296, 15178, 15106, 14992, 14868, 14738, 14593, 14438, 14283, 14169, 14064, 14004, 13914, 13824, 13752, 13671, 13590, 13515, 13458, 13380, 13305, 13230, 13143, 13025, 12935, 12878, 12794, 12743, 12656, 12596, 12521, 12443, 12359, 12278, 12200, 12131, 12047, 12002, 11948, 11891, 11828, 11744, 11663, 11588, 11495, 11402, 11288, 11204, 11126, 11039, 10961, 10883, 10787, 10679, 10583, 10481, 10360, 10227, 10113, 9961, 9828, 9717, 9584, 9485, 9324, 9112, 9019, 8908, 8766, 8584, 8426, 8211, 7920, 7663, 7406, 7152, 6904, 6677, 6453, 6265, 6101, 5904, 5716, 5489, 5307, 5056, 4850, 4569, 4284, 3966, 3712, 3518, 3342, 3206, 3048, 2909, 2773, 2668, 2596, 2512, 2370, 2295, 2232, 2166, 2103, 2022, 1956, 1887, 1830, 1803, 1770, 1728, 1674, 1635, 1599, 1557, 1500, 1482, 1434, 1389, 1356, 1317, 1284, 1245, 1200, 1179, 1140, 1110, 1092, 1062, 1044, 1035, 1014, 1008, 993, 981, 954, 936, 912, 894, 876, 864, 849, 828, 816, 801, 792, 777, 756, 732, 690, 660, 642, 615, 597, 576, 555, 522, 489, 459, 435, 411, 405, 396, 387, 375, 360, 354, 345, 344, 329, 314, 293, 278, 251, 236, 230, 224, 215, 214, 208, 199, 193, 184, 178, 169, 154, 127, 100, 94, 73, 37, 36, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26, 20, 19, 18, 17, 16, 15, 9, 0};

int cumf_TCOEF1[104]={16383, 13455, 12458, 12079, 11885, 11800, 11738, 11700, 11681, 11661, 11651, 11645, 11641, 10572, 10403, 10361, 10346, 10339, 10335, 9554, 9445, 9427, 9419, 9006, 8968, 8964, 8643, 8627, 8624, 8369, 8354, 8352, 8200, 8192, 8191, 8039, 8036, 7920, 7917, 7800, 7793, 7730, 7727, 7674, 7613, 7564, 7513, 7484, 7466, 7439, 7411, 7389, 7373, 7369, 7359, 7348, 7321, 7302, 7294, 5013, 4819, 4789, 4096, 4073, 3373, 3064, 2674, 2357, 2177, 1975, 1798, 1618, 1517, 1421, 1303, 1194, 1087, 1027, 960, 890, 819, 758, 707, 680, 656, 613, 566, 534, 505, 475, 465, 449, 430, 395, 358, 335, 324, 303, 295, 286, 272, 233, 215, 0};

int cumf_TCOEF2[104]={16383, 13582, 12709, 12402, 12262, 12188, 12150, 12131, 12125, 12117, 12113, 12108, 12104, 10567, 10180, 10070, 10019, 9998, 9987, 9158, 9037, 9010, 9005, 8404, 8323, 8312, 7813, 7743, 7726, 7394, 7366, 7364, 7076, 7062, 7060, 6810, 6797, 6614, 6602, 6459, 6454, 6304, 6303, 6200, 6121, 6059, 6012, 5973, 5928, 5893, 5871, 5847, 5823, 5809, 5796, 5781, 5771, 5763, 5752, 4754, 4654, 4631, 3934, 3873, 3477, 3095, 2758, 2502, 2257, 2054, 1869, 1715, 1599, 1431, 1305, 1174, 1059, 983, 901, 839, 777, 733, 683, 658, 606, 565, 526, 488, 456, 434, 408, 380, 361, 327, 310, 296, 267, 259, 249, 239, 230, 221, 214, 0};

int cumf_TCOEF3[104]={16383, 13532, 12677, 12342, 12195, 12112, 12059, 12034, 12020, 12008, 12003, 12002, 12001, 10586, 10297, 10224, 10202, 10195, 10191, 9223, 9046, 8999, 8987, 8275, 8148, 8113, 7552, 7483, 7468, 7066, 7003, 6989, 6671, 6642, 6631, 6359, 6327, 6114, 6103, 5929, 5918, 5792, 5785, 5672, 5580, 5507, 5461, 5414, 5382, 5354, 5330, 5312, 5288, 5273, 5261, 5247, 5235, 5227, 5219, 4357, 4277, 4272, 3847, 3819, 3455, 3119, 2829, 2550, 2313, 2104, 1881, 1711, 1565, 1366, 1219, 1068, 932, 866, 799, 750, 701, 662, 605, 559, 513, 471, 432, 403, 365, 336, 312, 290, 276, 266, 254, 240, 228, 223, 216, 206, 199, 192, 189, 0};

int cumf_TCOEFr[104]={16383, 13216, 12233, 11931, 11822, 11776, 11758, 11748, 11743, 11742, 11741, 11740, 11739, 10203, 9822, 9725, 9691, 9677, 9674, 8759, 8609, 8576, 8566, 7901, 7787, 7770, 7257, 7185, 7168, 6716, 6653, 6639, 6276, 6229, 6220, 5888, 5845, 5600, 5567, 5348, 5327, 5160, 5142, 5004, 4900, 4798, 4743, 4708, 4685, 4658, 4641, 4622, 4610, 4598, 4589, 4582, 4578, 4570, 4566, 3824, 3757, 3748, 3360, 3338, 3068, 2835, 2592, 2359, 2179, 1984, 1804, 1614, 1445, 1234, 1068, 870, 739, 668, 616, 566, 532, 489, 453, 426, 385, 357, 335, 316, 297, 283, 274, 266, 259, 251, 241, 233, 226, 222, 217, 214, 211, 209, 208, 0};

int cumf_TCOEF1_intra[104]={16383, 13383, 11498, 10201, 9207, 8528, 8099, 7768, 7546, 7368, 7167, 6994, 6869, 6005, 5474, 5220, 5084, 4964, 4862, 4672, 4591, 4570, 4543, 4397, 4337, 4326, 4272, 4240, 4239, 4212, 4196, 4185, 4158, 4157, 4156, 4140, 4139, 4138, 4137, 4136, 4125, 4124, 4123, 4112, 4111, 4110, 4109, 4108, 4107, 4106, 4105, 4104, 4103, 4102, 4101, 4100, 4099, 4098, 4097, 3043, 2897, 2843, 1974, 1790, 1677, 1552, 1416, 1379, 1331, 1288, 1251, 1250, 1249, 1248, 1247, 1236, 1225, 1224, 1223, 1212, 1201, 1200, 1199, 1198, 1197, 1196, 1195, 1194, 1193, 1192, 1191, 1190, 1189, 1188, 1187, 1186, 1185, 1184, 1183, 1182, 1181, 1180, 1179, 0};

int cumf_TCOEF2_intra[104]={16383, 13242, 11417, 10134, 9254, 8507, 8012, 7556, 7273, 7062, 6924, 6839, 6741, 6108, 5851, 5785, 5719, 5687, 5655, 5028, 4917, 4864, 4845, 4416, 4159, 4074, 3903, 3871, 3870, 3765, 3752, 3751, 3659, 3606, 3580, 3541, 3540, 3514, 3495, 3494, 3493, 3474, 3473, 3441, 3440, 3439, 3438, 3425, 3424, 3423, 3422, 3421, 3420, 3401, 3400, 3399, 3398, 3397, 3396, 2530, 2419, 2360, 2241, 2228, 2017, 1687, 1576, 1478, 1320, 1281, 1242, 1229, 1197, 1178, 1152, 1133, 1114, 1101, 1088, 1087, 1086, 1085, 1072, 1071, 1070, 1069, 1068, 1067, 1066, 1065, 1064, 1063, 1062, 1061, 1060, 1059, 1058, 1057, 1056, 1055, 1054, 1053, 1052, 0};

int cumf_TCOEF3_intra[104]={16383, 12741, 10950, 10071, 9493, 9008, 8685, 8516, 8385, 8239, 8209, 8179, 8141, 6628, 5980, 5634, 5503, 5396, 5327, 4857, 4642, 4550, 4481, 4235, 4166, 4151, 3967, 3922, 3907, 3676, 3500, 3324, 3247, 3246, 3245, 3183, 3168, 3084, 3069, 3031, 3030, 3029, 3014, 3013, 2990, 2975, 2974, 2973, 2958, 2943, 2928, 2927, 2926, 2925, 2924, 2923, 2922, 2921, 2920, 2397, 2298, 2283, 1891, 1799, 1591, 1445, 1338, 1145, 1068, 1006, 791, 768, 661, 631, 630, 615, 592, 577, 576, 561, 546, 523, 508, 493, 492, 491, 476, 475, 474, 473, 472, 471, 470, 469, 468, 453, 452, 451, 450, 449, 448, 447, 446, 0};

int cumf_TCOEFr_intra[104]={16383, 12514, 10776, 9969, 9579, 9306, 9168, 9082, 9032, 9000, 8981, 8962, 8952, 7630, 7212, 7053, 6992, 6961, 6940, 6195, 5988, 5948, 5923, 5370, 5244, 5210, 4854, 4762, 4740, 4384, 4300, 4288, 4020, 3968, 3964, 3752, 3668, 3511, 3483, 3354, 3322, 3205, 3183, 3108, 3046, 2999, 2981, 2974, 2968, 2961, 2955, 2949, 2943, 2942, 2939, 2935, 2934, 2933, 2929, 2270, 2178, 2162, 1959, 1946, 1780, 1651, 1524, 1400, 1289, 1133, 1037, 942, 849, 763, 711, 591, 521, 503, 496, 474, 461, 449, 442, 436, 426, 417, 407, 394, 387, 377, 373, 370, 367, 366, 365, 364, 363, 362, 358, 355, 352, 351, 350, 0};

int cumf_SIGN[3]={16383, 8416, 0};

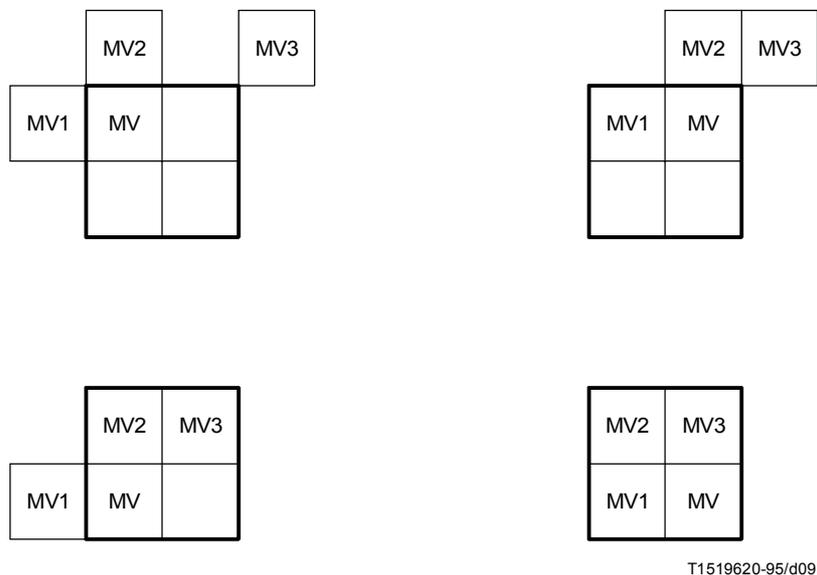
int cumf_LAST[3]={16383, 9469, 0};

int cumf_LAST_intra[3]={16383, 2820, 0};

int cumf_RUN[65]={16383, 15310, 14702, 13022, 11883, 11234, 10612, 10192, 9516, 9016, 8623, 8366, 7595, 7068, 6730, 6487, 6379, 6285, 6177, 6150, 6083, 5989, 5949, 5922, 5895, 5828, 5774, 5773, 5394, 5164, 5016, 4569, 4366, 4136, 4015, 3867, 3773, 3692, 3611, 3476, 3341, 3301, 2787, 2503, 2219, 1989, 1515, 1095, 934, 799, 691, 583, 435, 300, 246, 206, 125, 124, 97, 57, 30, 3, 2, 1, 0};

int cumf_RUN_intra[65]={16383, 10884, 8242, 7124, 5173, 4745, 4246, 3984, 3034, 2749, 2607, 2298, 966, 681, 396, 349, 302, 255, 254, 253, 206, 159, 158, 157, 156, 155, 154, 153, 106, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0};

Los vectores se obtienen sumando predictores a las diferencias de vectores indicadas por MVD y $MVD_{2,4}$ de manera similar a cuando sólo está presente un vector de movimiento por macrobloque, según las reglas de decisión dadas en 6.1.1. Los predictores se calculan por separado para los componentes horizontal y vertical. Pero los predictores candidatos MV1, MV2 y MV3 se vuelven a definir, como se indica en la Figura F.1. Si sólo está presente un vector por macrobloque, MV1, MV2 y MV3 se definen como para el bloque 8 * 8 numerado 1 en la Figura 5 (esta definición se muestra en la parte superior izquierda de las cuatro subfiguras de la Figura F.1).



T1519620-95/d09

FIGURA F.1/H.263

Redefinición de los predictores candidatos MV1, MV2 y MV3 para cada uno de los bloques de luminancia de un macrobloque

Cuando se utilizan cuatro vectores, cada uno de los vectores de movimiento se utiliza para todos los pixels de uno de los cuatro bloques de luminancia del macrobloque. La numeración de los vectores de movimiento es equivalente a la numeración de los cuatro bloques de luminancia, como se indica en la Figura 5. El vector de movimiento MVD_{CHR} para los dos bloques de crominancia se obtiene calculando la suma de los cuatro vectores de luminancia y dividiéndola por 8; los valores de componentes de los vectores resultantes con resolución de dieciseisavo de pixel se modifican hacia la posición de mitad de pixel más cercana, como se indica en el Cuadro F.1.

CUADRO F.1/H.263

Modificación de los componentes de vectores de crominancia con resolución de dieciseisavo de pixel

Posición de dieciseisavo de pixel	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	/16
Posición resultante	0	0	0	1	1	1	1	1	1	1	1	1	1	1	2	2	/2

Los valores de mitad de pixel se calculan por interpolación bilineal como se describe en 6.1.2. En el modo con predicción avanzada, la predicción para la luminancia se obtiene por compensación de movimiento de bloques superpuestos como se describe en F.3. La predicción para la crominancia se obtiene aplicando el vector de movimiento MVD_{CHR} a todos los pixels en los dos bloques de crominancia (como se hace en el modo con predicción por defecto).

F.3 Compensación de movimiento para bloques de luminancia superpuestos

Cada pixel de un bloque de predicción de luminancia de $8 * 8$ es una suma ponderada de tres valores de predicción, dividida por 8 (con redondeo). Para obtener los tres valores de predicción se utilizan tres vectores de movimiento: el vector de movimiento del bloque de luminancia elegido y dos de los cuatro vectores «distantes»:

- el vector de movimiento del bloque situado a la izquierda o a la derecha del bloque de luminancia elegido;
- el vector de movimiento del bloque situado por encima o por debajo del bloque de luminancia elegido.

Los vectores de movimiento distantes de otros GOB se utilizan de la misma manera que los vectores de movimiento distantes dentro del GOB elegido.

Para cada pixel, se utilizan los vectores de movimiento distantes de los bloques situados en el borde de los dos bloques más cercanos. Esto significa que, para la mitad superior del bloque, se usa el vector de movimiento correspondiente al bloque situado por encima del bloque elegido, mientras que para la mitad inferior del bloque, se utiliza el vector de movimiento correspondiente al bloque situado por debajo del bloque elegido (véase la Figura F.3). Análogamente, para la mitad izquierda del bloque, se utiliza el vector de movimiento correspondiente al bloque situado a la izquierda del bloque elegido, mientras que para la mitad derecha del bloque, se utiliza el vector de movimiento correspondiente al bloque situado a la derecha del bloque elegido (véase la Figura F.4).

La creación de cada pixel, $\bar{p}(i, j)$, es un bloque de predicción de luminancia $8 * 8$ que está regida por la siguiente ecuación:

$$\bar{p}(i, j) = (q(i, j) \times H_0(i, j) + r(i, j) \times H_1(i, j) + s(i, j) \times H_2(i, j) + 4) / 8$$

donde $q(i, j)$, $r(i, j)$ y $s(i, j)$ son los pixeles de la imagen referenciada definida por:

$$q(i, j) = p(i + MV^{0,x}, j + MV^{0,y})$$

$$r(i, j) = p(i + MV^{1,x}, j + MV^{1,y})$$

$$s(i, j) = p(i + MV^{2,x}, j + MV^{2,y})$$

En este caso, $(MV^{0,x}, MV^{0,y})$ denota el vector de movimiento para el bloque vigente. $(MV^{1,x}, MV^{1,y})$ denota el vector de movimiento del bloque por encima o por debajo y $(MV^{2,x}, MV^{2,y})$ denota el vector de movimiento a la izquierda o a la derecha del bloque vigente definido anteriormente.

Las matrices $H_0(i, j)$, $H_1(i, j)$ y $H_2(i, j)$ se definen en las Figuras F.2, F.3 y F.4, donde (i, j) indican respectivamente la columna y la fila de la matriz.

Si no se modificó uno de los macrobloques adyacentes, el vector de movimiento distante correspondiente se pone a 0. Si uno de los bloques adyacentes se codificó INTRA, el vector de movimiento distante correspondiente se sustituye por el vector de movimiento para el bloque vigente salvo cuando se utiliza el modo trama PB. En este caso (bloque INTRA en modo trama PB), se utiliza el vector de movimiento del bloque INTRA (véase también el Anexo G). Cuando el bloque vigente está en el borde de la imagen y, por lo tanto, falta un bloque colindante, el vector de movimiento distante correspondiente se sustituye por el vector de movimiento elegido. En todos los casos, si el bloque se encuentra en la parte más baja del macrobloque (para los bloques números 3 ó 4, véase la Figura 5), el vector de movimiento distante que corresponde al bloque de luminancia $8 * 8$ del macrobloque situado por debajo del macrobloque elegido se sustituye por el vector de movimiento del bloque elegido.

En las Figuras F.2, F.3 y F.4 se ofrecen los valores de ponderación de la predicción.

4	5	5	5	5	5	5	4
5	5	5	5	5	5	5	5
5	5	6	6	6	6	5	5
5	5	6	6	6	6	5	5
5	5	6	6	6	6	5	5
5	5	6	6	6	6	5	5
5	5	5	5	5	5	5	5
4	5	5	5	5	5	5	4

FIGURA F.2/H.263

Valores de ponderación, H_0 , para la predicción con el vector de movimiento del bloque de luminancia elegido

2	2	2	2	2	2	2	2
1	1	2	2	2	2	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	2	2	2	2	1	1
2	2	2	2	2	2	2	2

FIGURA F.3/H.263

Valores de ponderación, H_1 , para la predicción con los vectores de movimiento de los bloques de luminancia situados por encima y por debajo del bloque de luminancia elegido

2	1	1	1	1	1	1	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	1	1	1	1	1	1	2

FIGURA F.4/H.263

Valores de ponderación, H_2 , para la predicción con los vectores de movimiento de los bloques de luminancia situados a la izquierda y a la derecha del bloque de luminancia elegido

Anexo G

Modo trama PB

(Este anexo es parte integrante de esta Recomendación)

G.1 Introducción

Este anexo describe el modo trama PB facultativo de la Recomendación H.263. La capacidad de este modo se señala por medios externos (por ejemplo, aplicando la Recomendación H.245). El uso de este modo se indica en PTYPE.

Una trama PB consiste en dos imágenes que se codifican como una unidad. El nombre «PB» proviene del nombre de los tipos de imagen de la Recomendación H.262, en la que hay imágenes P e imágenes B. Por consiguiente, una trama PB consiste en una imagen P, que se predice a partir de la anterior imagen P decodificada, y una imagen B, que se predice a partir de la última imagen P decodificada y la imagen P que está siendo decodificada en ese momento. Se eligió la denominación imagen B porque partes de imágenes B se pueden predecir bidireccionalmente a partir de imágenes P pasadas y futuras. El proceso de predicción se ilustra en la Figura G.1.

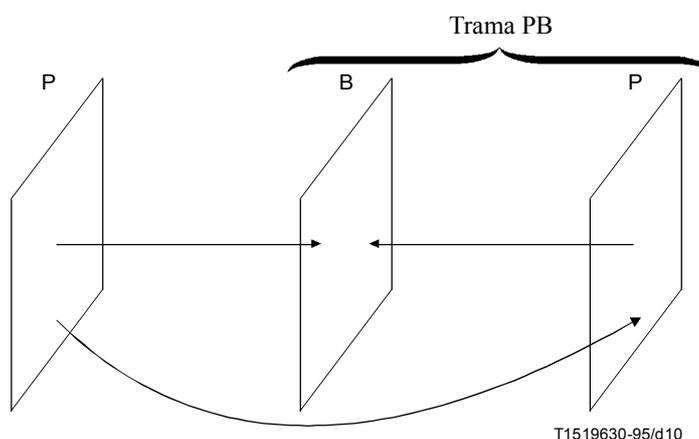


FIGURA G.1/H.263

Predicción con el modo trama PB

G.2 Tramas PB y bloques INTRA

Cuando se utilizan tramas PB, el modo de codificación INTRA tiene el significado siguiente (véase también 5.3.2):

- los bloques P tienen codificación INTRA;
- los bloques B tienen codificación INTER con la predicción correspondiente de bloque INTER.

Si se utilizan tramas PB, los datos de vectores de movimiento (MVD, *motion vector data*) se incluyen también para macrobloques INTRA en imágenes para las cuales PTYPE indica «INTER». En este caso, el vector se utiliza sólo para bloques B. Las palabras de código MVD_{2-4} no se utilizan nunca para INTRA (véase también el Cuadro 7). En el modo predicción avanzada y en el modo tramas PB y cuando uno de los bloques adyacentes se codificó en el modo INTRA, el correspondiente vector de movimiento distante no es sustituido por el vector de movimiento para el bloque urgente, sino que se usa el vector de movimiento «INTRA».

G.3 Capa de bloque

En el modo trama PB, un macrobloque consiste en doce bloques. Primero, se transmiten los datos de los seis bloques P, como en el modo H.263 por defecto; después, los datos de los seis bloques B (véase también 5.4). La Figura 10 muestra la estructura de la capa de bloque. INTRADC está presente en todos los bloques P del macrobloque si MCBPC indica tipo de MB 3 ó 4 (véanse el Cuadro 4 y el Cuadro 5). INTRADC no está presente en los bloques B. TCOEF está presente en los bloques P si así lo indican MCBPC o CBPY; TCOEF está presente en los bloques B si así lo indica CBPB.

G.4 Cálculo de los vectores de imagen B y con trama PB

Los vectores de imagen B se calculan como sigue (véase también 6.1.1). Supóngase un componente vectorial MV (vector de movimiento) de unidades de mitad de pixel para utilizar en la imagen P (MV representa un componente vectorial para un bloque de luminancia de $8 * 8$; si sólo se transmite un vector por macrobloque, MV tiene el mismo valor para cada uno de los cuatro bloques de luminancia de $8 * 8$). Para la predicción de una imagen B se necesitan los componentes vectoriales hacia adelante y hacia atrás, MV_F y MV_B . Estos componentes de vectores hacia adelante y hacia atrás se derivan de MV y a la larga son mejorados por un vector delta dado por MVDB.

- TR_D incremento de la referencia temporal TR del encabezamiento de la última imagen (véase 5.1.2). Si TR_D es negativo, $TR_D = TR_D + 256$.
- TR_B véase 5.1.7.

Supóngase que MV_D es el componente vectorial delta dado por MVDB y que corresponde con el componente vectorial MV. Si MVDB no está presente, MV_D se pone a cero. Si MVDB está presente, se utiliza el mismo MV_D dado por MVDB para cada uno de los cuatro bloques de luminancia B dentro del macrobloque.

Ahora MV_F y MV_B vienen dados en unidades de mitad de pixel por las siguientes fórmulas:

$$MV_F = (TR_B \times MV) / TR_D + MV_D$$

$$MV_B = ((TR_B - TR_D) \times MV) / TR_D \quad \text{si } MV_D \text{ es igual a } 0$$

$$MV_B = MV_F - MV \quad \text{si } MV_D \text{ es distinto de } 0$$

donde «/» significa división por truncamiento. Se supone que el ajuste por escalón refleja la posición real en el tiempo de las imágenes P y B. Se aprovecha el hecho de que la gama de valores de MV_F está limitada. Cada palabra VLC para MVDB representa un par de valores diferencia de los que sólo uno tendrá un valor para MV_F que cae dentro de la gama permitida $[-16; 15,5]$ en el modo vector de movimiento sin restricción $[-31,5; 31,5]$. También se utilizan las fórmulas de MV_F y MV_B en el caso de bloques INTRA, en los que se utilizan los datos de vector sólo para la predicción de bloques B.

Para los bloques de crominancia, MV_F se obtiene calculando la suma de los cuatro vectores de luminancia MV_F correspondientes y dividiéndola por 8; los componentes de vectores resultantes con resolución de dieciseisavo de pixel se modifican hacia la posición de mitad de pixel más cercana, como se indica en el Cuadro F.2. MV_B para crominancia, se obtiene calculando la suma de los cuatro vectores correspondientes de luminancia MV_B y dividiendo esta suma por 8; los componentes de vectores resultantes con resolución de un cuarto de pixel se modifican hacia la posición más cercana de mitad de pixel, como se indica en el Cuadro F.2.

Un valor positivo del componente horizontal o vertical del vector de movimiento significa que la predicción se forma a partir de pixels de la imagen P referenciada, situados a la derecha o debajo de los pixels objeto de predicción.

G.5 Predicción de un bloque B en una trama PB

En esta sección, un bloque significa un bloque de 8×8 . El procedimiento que se ofrece a continuación es aplicable a bloques de luminancia y de crominancia. Primero, se calculan los vectores hacia adelante y hacia atrás. Se supone que el macrobloque P (luminancia y crominancia) se decodifica, se reconstruye y se recorta (véase 6.3.2). Este macrobloque se denomina P_{REC} . Sobre la base de P_{REC} y la predicción de P_{REC} , se calcula la predicción del bloque B.

La predicción del bloque B tiene dos modos, que se usan en partes diferentes del bloque:

- Para los pixels en los que el vector hacia atrás – MV_B – apunta hacia adentro de P_{REC} , se utiliza la predicción bidireccional. Esto se obtiene como el promedio entre la predicción hacia adelante con MV_F relativo a la imagen P decodificada previamente y la predicción hacia atrás con MV_B relativo a P_{REC} . El promedio se calcula dividiendo la suma de las dos predicciones por dos (división por truncamiento).
- Para todos los demás pixels, se emplea la predicción hacia adelante con MV_F relativo a la imagen P decodificada previamente.

La Figura G.2 indica cuáles son las partes de un bloque que se predicen bidireccionalmente (partes sombreadas del bloque B) o sólo mediante predicción hacia adelante (resto del bloque B).

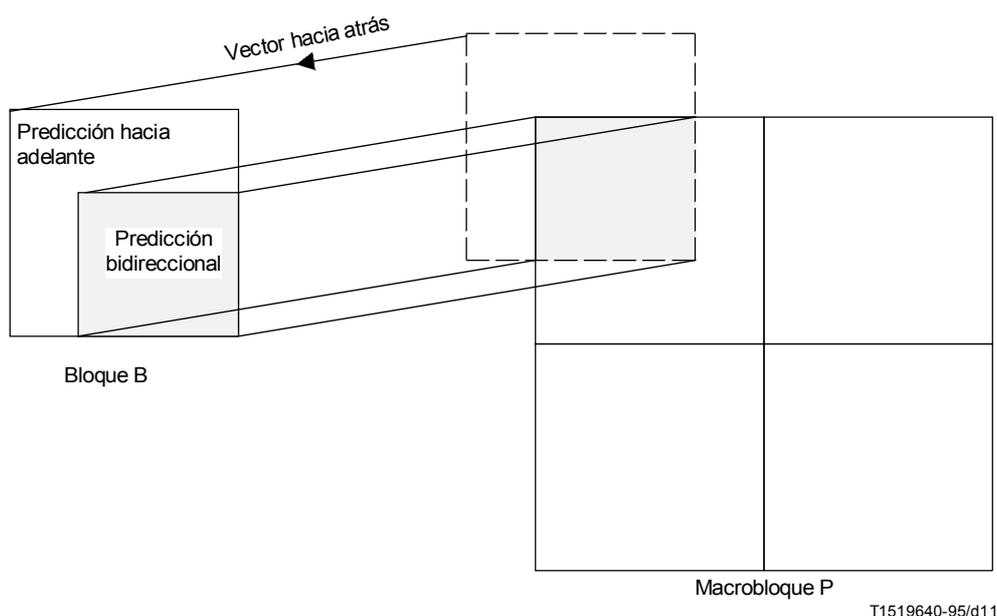


FIGURA G.2/H.263
Predicción hacia adelante y bidireccional para un bloque B

La predicción bidireccional se utiliza para los pixels en los que el vector hacia atrás – MV_B – apunta hacia adentro de P_{REC} . Estos pixels se definen mediante los procedimientos siguientes, especificados en C.

Definiciones:

- nh Posición horizontal del bloque en un macrobloque (0 ó 1).
- nv Posición vertical del bloque en un macrobloque (0 ó 1).
- mh(nh, nv) Componente vectorial horizontal del bloque (nh, nv) en unidades de mitad de pixel.
- mv(nh, nv) Componente vectorial vertical del bloque (nh, nv) en unidades de mitad de pixel.
- mhc Componente horizontal del vector de crominancia.
- mvc Componente vertical del vector de crominancia.

Procedimiento para la luminancia

```
for (nh = 0; nh <= 1; nh++) {
    for (nv = 0; nv <= 1; nv++) {
        for (i = nh * 8 + max(0,(-mh(nh,nv)+1)/2 - nh * 8);
            i <= nh * 8 + min(7,15-(mh(nh,nv)+1)/2 - nh * 8); i++) {
            for (j = nv * 8 + max(0,(-mv(nh,nv)+1)/2 - nv * 8);
                j <= nv * 8 + min(7,15-(mv(nh,nv)+1)/2 - nv * 8); j++) {
                predict pixel (i,j) bidirectionally
            }
        }
    }
}
```

Procedimiento para la crominancia

```
for (i = max(0,(-mhc+1)/2); i <= min(7,7-(mhc+1)/2); i++) {
    for (j = max(0,(-mvc+1)/2); j <= min(7,7-(mvc+1)/2); j++) {
        predict pixel (i,j) bidirectionally;
    }
}
```

Los pixels que no se predicen bidireccionalmente se predicen sólo con predicción hacia adelante.

Anexo H

Corrección de errores hacia adelante para señales de vídeo codificadas

(Este anexo es parte integrante de esta Recomendación)

H.1 Introducción

En este anexo se describe un método de corrección de errores hacia adelante facultativo (código y alineación de trama) para la transmisión de datos de vídeo codificados H.263. Esta corrección de errores hacia adelante se puede emplear en situaciones en las que no se proporciona corrección de errores hacia adelante por medios externos, por ejemplo a nivel de multiplex o de sistema. No se utiliza para la Recomendación H.324. La alineación de trama y el código de corrección de errores hacia adelante son iguales a los de la Recomendación H.261.

H.2 Alineación de trama para la corrección de errores

Para que el decodificador pueda identificar la información de paridad de los datos de vídeo y de corrección de errores, se incluye un esquema de alineación de trama para la corrección de errores. Este esquema consiste en multitramas de 8 tramas, cada una de las cuales consta de 1 bit de alineación de trama, 1 bit indicador de relleno (Fi), 492 bits de datos codificados (o de relleno, todos «1») y 18 bits de paridad (véase la Figura H.1). Para cada multitrama, el esquema de alineación de trama constituido por los bits de alineación de trama de las 8 tramas individuales es:

$$(S_1S_2S_3S_4S_5S_6S_7S_8) = (00011011)$$

El indicador de relleno (Fi) puede ser puesto a 0 por un codificador. En este caso, se utilizan 492 bits de relleno consecutivos (todos «1») en lugar de 492 bits de datos codificados. Esto se puede usar para datos de relleno (véase 3.6).

H.3 Código de corrección de errores

El código de corrección de errores es un código de corrección de errores hacia adelante BCH (511,493); su uso por el decodificador es opcional. La paridad se calcula con un código de 493 bits, que consta de un indicador de relleno (Fi) de 1 bit y 492 bits de datos de vídeo codificados.

El polinomio generador es:

$$g(x) = (x^9 + x^4 + 1)(x^9 + x^6 + x^4 + x^3 + 1)$$

Ejemplo: Para los datos de entrada «01111 ... 11» (493 bits), los bits resultantes de paridad de corrección son «011011010100011011» (18 bits).

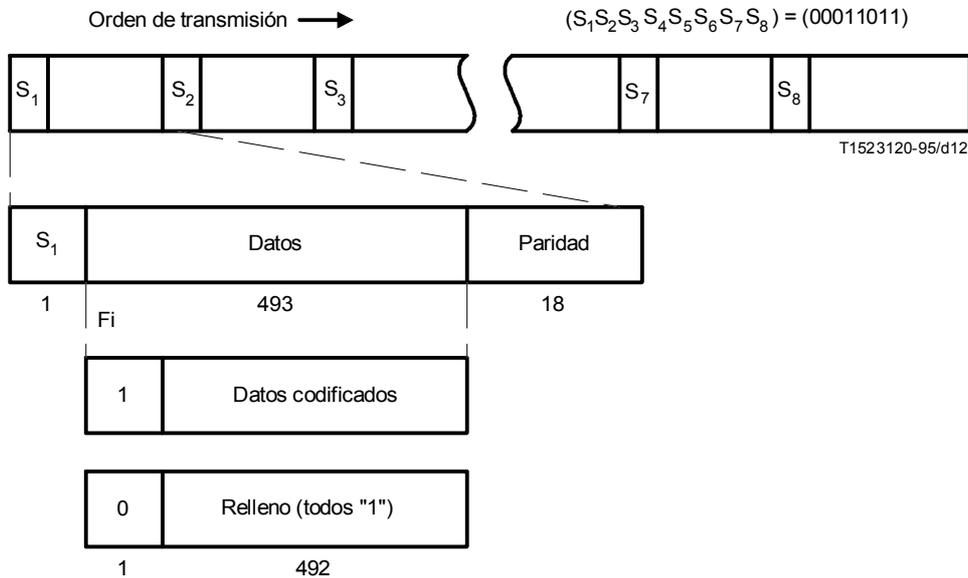


FIGURA H.1/H.263
Trama de corrección de errores

H.4 Plazo de reenganche para la alineación del corrector de errores

No se puede considerar alcanzado el reenganche de trama antes de recibir tres esquemas consecutivos de alineación de trama para la corrección de errores. El decodificador debe estar diseñado de manera tal que restablezca el enganche de trama dentro de los 34 000 bits que siguen a un cambio de fase de la alineación de trama del corrector de errores.

NOTA – Esto supone que los datos de vídeo no contienen tres emulaciones correctamente dispuestas en fase de la secuencia de alineación de trama para la corrección de errores durante el periodo de reconstitución.

SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie B	Medios de expresión
Serie C	Estadísticas generales de telecomunicaciones
Serie D	Principios generales de tarificación
Serie E	Red telefónica y RDSI
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión
Serie H	Transmisión de señales no telefónicas
Serie I	Red digital de servicios integrados (RDSI)
Serie J	Transmisiones de señales radiofónicas y de televisión
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
Serie M	Mantenimiento: sistemas de transmisión, circuitos telefónicos, telegrafía, facsimil y circuitos arrendados internacionales
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Calidad de transmisión telefónica
Serie Q	Conmutación y señalización
Serie R	Transmisión telegráfica
Serie S	Equipos terminales de telegrafía alfabética
Serie T	Equipos terminales y protocolos para los servicios de telemática
Serie U	Conmutación telegráfica
Serie V	Comunicación de datos por la red telefónica
Serie X	Redes de comunicación de datos y comunicación entre sistemas abiertos
Serie Z	Lenguajes de programación



Impreso en Suiza

Ginebra, 1996