



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

T.82

(03/93)

**EQUIPOS TERMINALES Y PROTOCOLOS
PARA LOS SERVICIOS DE TELEMÁTICA**

**TECNOLOGÍA DE LA INFORMACIÓN –
REPRESENTACIÓN CODIFICADA DE
INFORMACIÓN DE IMAGEN Y DE AUDIO –
COMPRESIÓN DE IMAGEN BINIVEL
PROGRESIVA**

Recomendación UIT-T T.82

(Anteriormente «Recomendación del CCITT»)

Prefacio

La UIT (Unión Internacional de Telecomunicaciones) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones. El UIT-T (Sector de Normalización de las Telecomunicaciones de la UIT) es un órgano permanente de la UIT. En el UIT-T, que es la entidad que establece normas mundiales (Recomendaciones) sobre las telecomunicaciones, participan unos 179 países miembros, 84 empresas de explotación de telecomunicaciones, 145 organizaciones científicas e industriales y 38 organizaciones internacionales.

Las Recomendaciones las aprueban los miembros del UIT-T de acuerdo con el procedimiento establecido en la Resolución N.º 1 de la CMNT (Helsinki, 1993). Adicionalmente, la Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), que se celebra cada cuatro años, aprueba las Recomendaciones que para ello se le sometan y establece el programa de estudios para el periodo siguiente.

En ciertos sectores de la tecnología de la información que corresponden a la esfera de competencia del UIT-T, se preparan las normas necesarias en colaboración con la ISO y la CEI. El texto de la Recomendación UIT-T T.82, se aprobó por la CMNT (Helsinki, 1-12 de marzo de 1993). Su texto se publica también, en forma idéntica, como Norma Internacional ISO/CEI 11544.

NOTAS

1 Como consecuencia de un proceso de reforma interna de la Unión Internacional de Telecomunicaciones (UIT), desde el 28 de febrero de 1993 ya no existe el CCITT. En su lugar, se creó el Sector de Normalización de las Telecomunicaciones de la UIT (UIT-T) con vigencia a partir del primero de marzo de 1993. De modo similar, en este proceso de reforma, el CCIR y la IFRB han sido sustituidos por el Sector de Radiocomunicaciones.

Con el fin de no retrasar la publicación de la presente Recomendación, no se han cambiado en el texto las referencias que contienen los acrónimos «CCITT, CCIR o IFRB» ni sus entidades relacionadas tales como Asamblea Plenaria, Secretaría, etc.. Las futuras ediciones de la presente Recomendación contendrán la terminología apropiada relacionada con la nueva estructura de la UIT.

2 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 1994

Es propiedad. Ninguna parte de esta publicación puede producirse 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>
Intro. 1	Características generales.....	i
Intro. 2	Barras y ordenación de datos.....	ii
Intro. 3	Bloques funcionales del codificador.....	iv
Intro. 3.1	Codificador de capa diferencial	v
Intro. 3.2	Codificador de capa de resolución mínima	vii
Intro. 4	Bloques funcionales del decodificador.....	viii
1	Alcance.....	1
2	Referencias normativas	1
3	Definiciones	1
4	Símbolos y siglas.....	3
4.1	Siglas.....	3
4.2	Constantes simbólicas	3
4.3	Símbolos matemáticos, operadores e indicadores.....	3
4.4	Variables con nombres mnemotécnicos.....	3
5	Convenios.....	4
5.1	Convenios y símbolos de los diagramas de flujo	4
5.2	Gráficos de plantilla.....	4
5.3	Fase espacial	5
5.4	Gráficos de estructura de datos	6
6	Requisitos.....	7
6.1	Reglas generales.....	7
6.2	Organización de los datos	8
6.3	Reducción de resolución	13
6.4	Predicción típica de capa diferencial.....	14
6.5	Predicción típica de capa de resolución mínima	16
6.6	Predicción determinística (DP)	20
6.7	Plantillas modelo y plantillas adaptativas	24
6.8	Codificación aritmética.....	27
7	Métodos de prueba y ejemplos de trenes de datos	44
7.1	Codificación aritmética.....	45
7.2	Algoritmo parametrizado	52
7.3	Ejemplos de tren de datos	56
Anexo A	– Soporte mínimo sugerido de los parámetros libres	57
Anexo B	– Diseño de la tabla de reducción de resolución	58
B.1	Filtrado.....	58
B.2	Excepciones	59
Anexo C	– Cambios de plantilla adaptativa	61
C.1	Generalidades.....	61
C.2	Capas diferenciales	61
C.3	Capa de resolución mínima.....	61

	<i>Página</i>
Anexo D – Diseño de la tabla de estimación de probabilidades.....	64
D.1 Estimación bayesiana.....	64
D.2 Múltiples contextos.....	64
D.3 Parametrización MPS/LPS.....	64
D.4 Seguimiento rápido.....	65
D.5 Reducción de la carga de cálculo.....	65
Anexo E – Patentes.....	69
E.1 Observaciones de introducción.....	69
E.2 Lista de patentes.....	69
E.3 Direcciones de contacto para informaciones sobre patentes.....	70
Anexo F – Bibliografía.....	72

Introducción y visión de conjunto

(Esta introducción no es parte integrante de esta Recomendación | Norma Internacional)

Esta Recomendación | Norma Internacional fue preparada por el Grupo mixto de expertos en imagen binivel (JBIG) de ISO/CEI JTC1/SC29/WG9 y la Comisión de Estudio VIII del CCITT. Este grupo de expertos se formó en 1988 con el fin de establecer una norma para la codificación progresiva de imágenes binivel.

Un sistema de codificación progresiva transmite una imagen comprimida enviando primero los datos comprimidos de una versión de resolución reducida de la imagen y potenciándolos luego en la medida necesaria con la transmisión de datos comprimidos adicionales, que se superponen a los ya transmitidos. Esta Recomendación | Norma Internacional define un método de codificación incluyendo los modos progresivo, secuencial compatible progresivo y secuencial simple, y sugiere un método para obtener las necesarias reproducciones a baja resolución. Se ha comprobado que es posible utilizar eficazmente los algoritmos de codificación y de reducción de resolución definidos para la codificación sin pérdidas de imágenes de escala de grises y en color, así como de imágenes binivel.

Las cláusulas de introducción y visión de conjunto y los Anexos A a F no forman parte integrante de la presente Recomendación | Norma Internacional.

Intro. 1 Características generales

Esta Especificación define un método para codificar por compresión, sin pérdida, una imagen binivel (es decir, una imagen que, como una en blanco y negro, tiene sólo dos colores). Dado que el método es adaptativo, puede también utilizarse eficazmente para codificar imágenes de la escala de grises y de color. Al ser adaptativo a las características de imagen, es robusto con relación al tipo de imagen. En las imágenes exploradas de caracteres impresos, las relaciones de compresión observadas han sido de 1,1 a 1,5 veces mayores que las obtenidas mediante el algoritmo de codificación MMR de las Recomendaciones T.4 (G3) y T.6 (G4) (que es menos complejo). En las imágenes de caracteres impresos generadas por computador, las relaciones de compresión observadas han sido hasta cinco veces mayores. En las imágenes con escala de grises reproducidas por generación de semitonos o por tremolación, se han observado relaciones de compresión de 2 a 30 veces mayores.

El método preserva los bits, lo que significa que, al igual que las Recomendaciones T.4 y T.6, no produce distorsión, y que la imagen decodificada final es idéntica a la original.

El método tiene también capacidad «progresiva». Al decodificar una imagen codificada progresivamente, se obtiene primero una reproducción de baja resolución de la imagen original con duplicaciones posteriores de la resolución a medida que se decodifican más datos. Obsérvese que la reducción de resolución se lleva a cabo de las capas de resolución superior hacia las capas de resolución inferior, mientras que la decodificación se lleva a cabo en el sentido inverso. La imagen de más baja resolución enviada en una secuencia progresiva es una imagen codificada secuencialmente. En una aplicación de codificación secuencial de una sola progresión, ésta es la única imagen enviada.

Las codificaciones progresivas tienen dos ventajas bien diferenciadas. La primera es que permiten diseñar una aplicación con una base de datos común que puede atender eficazmente dispositivos de salida cuyas capacidades de resolución son muy diferentes. Sólo es necesario enviar y decodificar la porción del fichero de la imagen comprimida que se necesita para la reconstrucción a la capacidad de resolución del dispositivo de salida en cuestión. Además, si se desea una potenciación adicional de la resolución, por ejemplo, para obtener una copia en papel de algo ya escrito en una pantalla de rayos catódicos (CRT), sólo hay que enviar la información necesaria para dicha potenciación.

La otra ventaja de las codificaciones progresivas es que pueden proporcionar una consulta rápida de las imágenes (en una pantalla de rayos catódicos) subjetivamente superior, para aplicaciones que utilizan enlaces de comunicaciones a velocidad baja y mediana. Se transmite y visualiza rápidamente una reproducción de baja resolución, seguida luego por la potenciación de resolución deseada. Cada fase de potenciación de la resolución se incorpora a la imagen ya disponible. La codificación progresiva facilita a un usuario el rápido reconocimiento de la imagen a medida que se visualiza, lo que a su vez le permite interrumpir la transmisión de la imagen.

D designa el número de duplicaciones de resolución (denominadas capas diferenciales) proporcionadas por la codificación progresiva. I_D designa la imagen de resolución máxima y X_D e Y_D sus dimensiones horizontal y vertical en pixels. R_D designa la resolución de muestreo de la imagen I_D .

Esta Especificación no impone casi ninguna restricción a los parámetros R_D , X_D , Y_D o D . Elecciones tales como 400 ó 200 puntos por pulgada (dpi, *dots-per-inch*) para la resolución R_D de la capa superior dan lugar a una jerarquía de resoluciones proporcionada a las actuales normas facsímil. Elegir una R_D de 600 ó 300 dpi produce una jerarquía progresiva más compatible con las resoluciones de las impresoras láser disponibles a partir de la redacción de esta Especificación.

Se prevé que D se elegirá normalmente de manera que la resolución mínima sea aproximadamente de 10 a 25 dpi. Las imágenes binivel típicas no son legibles cuando se reducen a dicha resolución, pero no obstante estas reproducciones de baja resolución siguen siendo bastante útiles y funcionan como iconos automáticamente generados. La presentación en página suele ser de tipo evidente, y a menudo es posible el reconocimiento de determinadas páginas que se han visto antes a resolución superior.

Como antes mencionado, esta Especificación no limita el número D de duplicaciones de resolución. Puede ponerse a 0 si la codificación progresiva no presenta ninguna utilidad, como es el caso, por ejemplo, para el facsímil impreso. Esto preserva la ventaja de la JBIG en materia de compresión con relación a la MMR (y, de hecho, generalmente la aumenta en cierta medida), a la vez que elimina la necesidad de las memorias intermedias y simplifica el algoritmo. La codificación JBIG secuencial de una sola progresión tiene aplicaciones potenciales idénticas a las de la codificación MMR. Las imágenes comprimidas por un codificador secuencial de una sola progresión serán legibles por decodificadores capaces de decodificar progresivamente, si bien únicamente la versión de más baja resolución de una imagen codificada progresivamente será decodificable por un decodificador secuencial de una sola progresión.

Es posible utilizar esta Especificación para la codificación sin pérdidas de imágenes en escala de grises y en color codificando los planos de bits independientemente, como si cada uno fuera en sí mismo una imagen binivel. Este método de codificación de imágenes en escala de grises o en color puede ser utilizado como alternativa a la norma de codificación fotográfica de la Rec. T.81 del CCITT | ISO/CEI 10918-1 (JPEG) en su modo sin pérdidas. Resultados experimentales preliminares han mostrado que la JBIG tiene una ventaja de compresión sobre el JPEG en su modo sin pérdidas para las imágenes en escala de grises de hasta 6 bits por pixel. Para 6 a 8 bits por pixel, los resultados de la compresión han sido similares para ambos métodos. Esta Especificación abarca las imágenes con más de un plano de bits, pero no formula ninguna recomendación sobre la manera de hacer corresponder las intensidades de la escala de grises o de color con los planos de bits. Experimentalmente, se ha descubierto que para imágenes en escala de grises una correspondencia por codificación Gray de la intensidad es superior a una correspondencia por simple codificación binaria ponderada de la intensidad.

Intro. 2 Barras y ordenación de datos

Cuando es necesario distinguir la codificación progresiva de la forma más tradicional de codificación de imagen, en la que la imagen se codifica a resolución completa de izquierda a derecha y de arriba a abajo, esta forma más antigua de codificación se denominará «secuencial». La ventaja de la codificación secuencial sobre la codificación progresiva es que no se requiere memoria intermedia de trama. La codificación progresiva requiere una memoria intermedia de trama en su resolución siguiente a la máxima, dado que se utilizan imágenes de resolución inferior al codificar imágenes de resolución más alta.

Es posible crear un tren de datos JBIG con sólo una capa de resolución mínima, lo que puede denominarse codificación secuencial de una sola progresión. En dicha codificación, una imagen de resolución completa es codificada sin referencia a ninguna capa de resolución diferencial. El parámetro D (mencionado en Intro. 1) se pone a cero. Debe observarse que en la codificación progresiva de una imagen la capa de resolución mínima es de hecho codificada en codificación secuencial de una sola progresión. Si se codifica una imagen de resolución completa mediante la codificación secuencial de una sola progresión, no será posible decodificar la imagen progresivamente.

Se dice que la codificación en el modo secuencial compatible progresivo es «compatible» con la codificación en el modo progresivo, ya que los trenes de datos creados (codificador) o leídos (decodificador) en cualquiera de los modos transportan exactamente la misma información. Todo lo que cambia al pasar de la codificación progresiva a la codificación secuencial compatible progresiva es el orden en el que las partes del mensaje son creadas por el codificador. Todo lo que cambia al pasar de la decodificación progresiva a la decodificación secuencial compatible progresiva es el orden en que estas partes son utilizadas por el decodificador.

Esta compatibilidad se obtiene descomponiendo una imagen en partes más pequeñas antes de la compresión. Estas partes son creadas dividiendo la imagen en cada una de sus «capas» de resolución en bandas horizontales denominadas «barras». La codificación secuencial compatible progresiva requiere una memoria intermedia «de barra» (mucho más pequeña que una memoria intermedia de página) y cierta memoria adicional utilizada para la codificación de entropía adaptativa de cada capa de resolución.

La Figura Intro. 1 ilustra dicha descomposición cuando hay tres capas de resolución, tres barras por capa y un solo plano de bits. El Cuadro Intro. 1 muestra formas definidas de secuenciar a nueve barras.

Obsérvese que además de la distinción entre progresiva y secuencial que transporta el bit SEQ, existe también una distinción de orden de resolución que transporta el bit HITOLO. Los codificadores trabajan desde alta resolución hacia abajo, por lo que codifican de lo más naturalmente las barras en orden HITOLO. Los decodificadores deben componer la imagen desde una resolución baja, por lo que procesan de lo más naturalmente las barras en sentido opuesto. Cuando un codificador envía datos directamente a un decodificador, uno u otro deben retener en memoria intermedia para invertir el orden. Cuando una aplicación incluye una base de datos, ésta (con las asignaciones apropiadas) puede ser utilizada para la retención en memoria intermedia y la inversión del orden (incluida la asignación de un valor apropiado a HITOLO), eliminando así esta necesidad en el codificador y el decodificador.

Una barra tiene un tamaño vertical que suele ser mucho menor que el de la imagen entera. El número L_0 de líneas por barra en la capa más baja es otro parámetro libre. Por ejemplo, L_0 puede elegirse de manera que una barra tenga unos 8 mm. De hacerse esta elección, el número S de barras de una imagen de una hoja de papel de tamaño carta comercial será de unas 35.

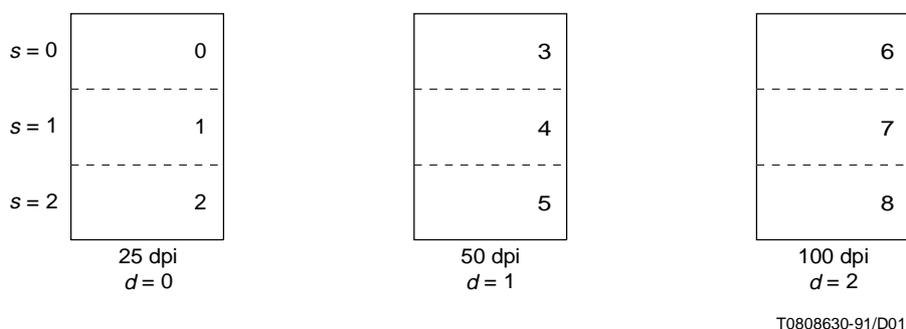


Figura Intro. 1 – Descomposición en el caso especial de tres capas, tres barras y un plano de bits

Cuadro Intro. 1 – Posibles ordenaciones de datos binivel

HITOLO	SEQ	Ejemplo de orden
0	0	0, 1, 2 3, 4, 5 6, 7, 8
0	1	0, 3, 6 1, 4, 7 2, 5, 8
1	0	6, 7, 8 3, 4, 5 0, 1, 2
1	1	6, 3, 0 7, 4, 1 8, 5, 2

Cuando hay más de un plano de bits, como en la Figura Intro. 2, existen doce ordenaciones de barra definidas. El Cuadro Intro. 2 las enumera. Como antes, el bit HITOLO transporta la distinción de orden de resolución, y el bit SEQ lleva la distinción entre progresiva y secuencial. El bit ILEAVE puesto a 1 indica el entrelazado de múltiples planos de bits. El bit SMID puesto a 1 indica que s , el índice en la barra, está en la mitad, como se muestra más claramente en el Cuadro 11 de 6.2.4.

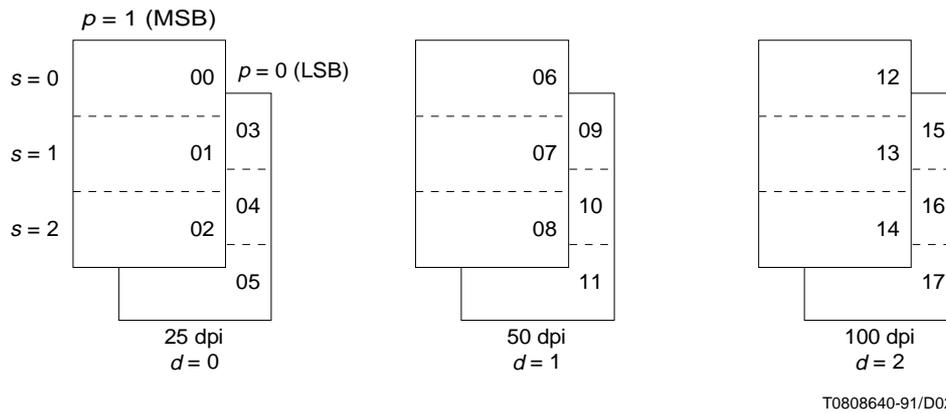


Figura Intro. 2 – Descomposición del caso especial de tres capas, tres barras y dos planos de bits

Cuadro Intro. 2 – Posibles ordenaciones de datos multiplano

HITOLLO	SEQ	ILEAVE	SMID	Ejemplo de orden
0	0	0	0	(00,01,02 06,07,08 12,13,14) (03,04,05 09,10,11 15,16,17)
0	0	1	0	(00,01,02 03,04,05) (06,07,08 09,10,11) (12,13,14 15,16,17)
0	0	1	1	(00,03 01,04 02,05) (06,09 07,10 08,11) (12,15 13,16 14,17)
0	1	0	0	(00,06,12 03,09,15) (01,07,13 04,10,16) (02,08,14 05,11,17)
0	1	0	1	(00,06,12 01,07,13 02,08,14) (03,09,15 04,10,16 05,11,17)
0	1	1	0	(00,03 06,09 12,15) (01,04 07,10 13,16) (02,05 08,11 14,17)
1	0	0	0	(12,13,14 06,07,08 00,01,02) (15,16,17 09,10,11 03,04,05)
1	0	1	0	(12,13,14 15,16,17) (06,07,08 09,10,11) (00,01,02 03,04,05)
1	0	1	1	(12,15 13,16 14,17) (06,09 07,10 08,11) (00,03 01,04 02,05)
1	1	0	0	(12,06,00 15,09,03) (13,07,01 16,10,04) (14,08,02 17,11,05)
1	1	0	1	(12,06,00 13,07,01 14,08,02) (15,09,03 16,10,04 17,11,05)
1	1	1	0	(12,15 06,09 00,03) (13,16 07,10 01,04) (14,17 08,11 02,05)

Las dos nuevas variables ILEAVE y SMID, con las dos variables anteriores HITOLO y SEQ, hacen posible indexar estos doce órdenes. Cuatro de las dieciseis posibles combinaciones de estas cuatro variables binarias no tienen ninguna ordenación de barras asociada con ellas. Si existe sólo un plano, el orden de barras no depende de ILEAVE y SMID, y sus valores son inconsecuentes.

Los datos comprimidos $C_{s,d,p}$ para la barra s de la capa de resolución d del plano de bits p son independientes de la ordenación de barras. Lo único que cambia a medida que varían HITOLO, SEQ, ILEAVE y SMID es el orden en el que los datos se concatenan en un tren de datos. Esta es la característica compatible antes observada.

Para mayor sencillez, en el resto de este punto introductorio se supondrá que existe sólo un plano de bits, y el subíndice p que denota el plano de bits se suprimirá de $C_{s,d,p}$.

Intro. 3 Bloques funcionales del codificador

Un codificador puede descomponerse como se muestra en la Figura Intro. 3. (En codificación secuencial de una sola progresión se utilizaría únicamente el codificador de la capa de resolución mínima.)

Aunque teóricamente hay D codificadores de capa diferencial, como se muestra en la Figura Intro. 3, algunas realizaciones pueden optar por utilizar recursivamente un codificador de capa diferencial física.

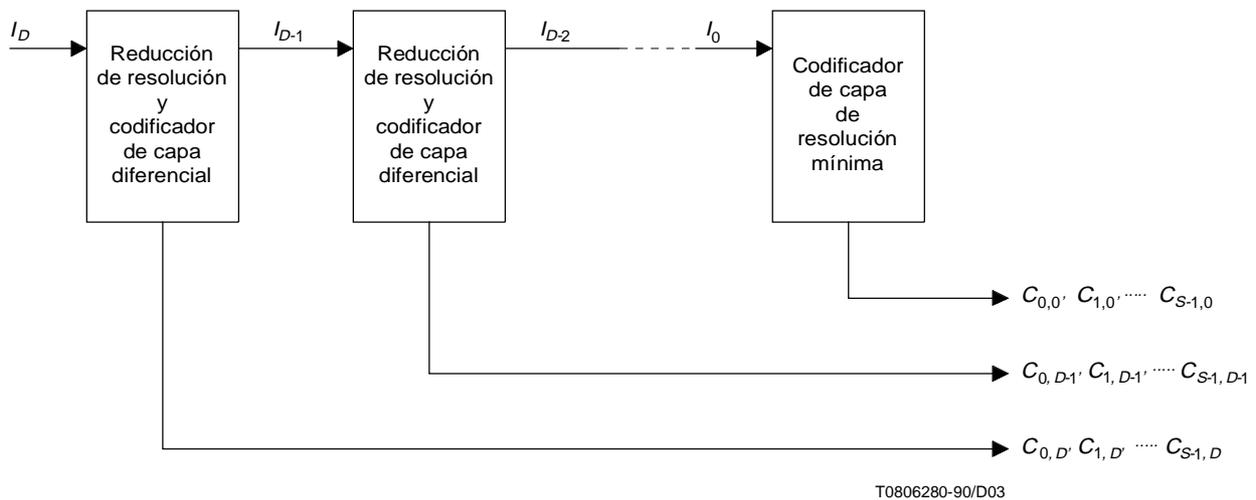


Figura Intro. 3 – Descomposición del codificador

Intro. 3.1 Reducción de resolución y codificador de capa diferencial

Cada uno de los bloques de reducción de resolución y codificador de capa diferencial de la Figura Intro. 3 es idéntico en función, por lo que sólo se necesita una descripción de la operación en una capa. En dicha descripción intervienen sólo dos capas de resolución. Por sencillez, en el resto de esta subcláusula, la imagen entrante se denominará imagen de «alta resolución» y la saliente, imagen de «baja resolución». Se señala, no obstante, que las imágenes de «alta» y «baja» resolución de cualquier bloque de capa diferencial de la Figura Intro. 3 no son en general las imágenes de resolución máxima y mínima de todo el sistema.

Un bloque de codificación de capa diferencial de la Figura Intro. 3 puede a su vez descomponerse en subbloques, como se muestra en la Figura Intro. 4. No es necesario utilizar todos los bloques en todos los sistemas. Para la definición de los nombres de señales, refiérase a los cuadros de la cláusula 4.

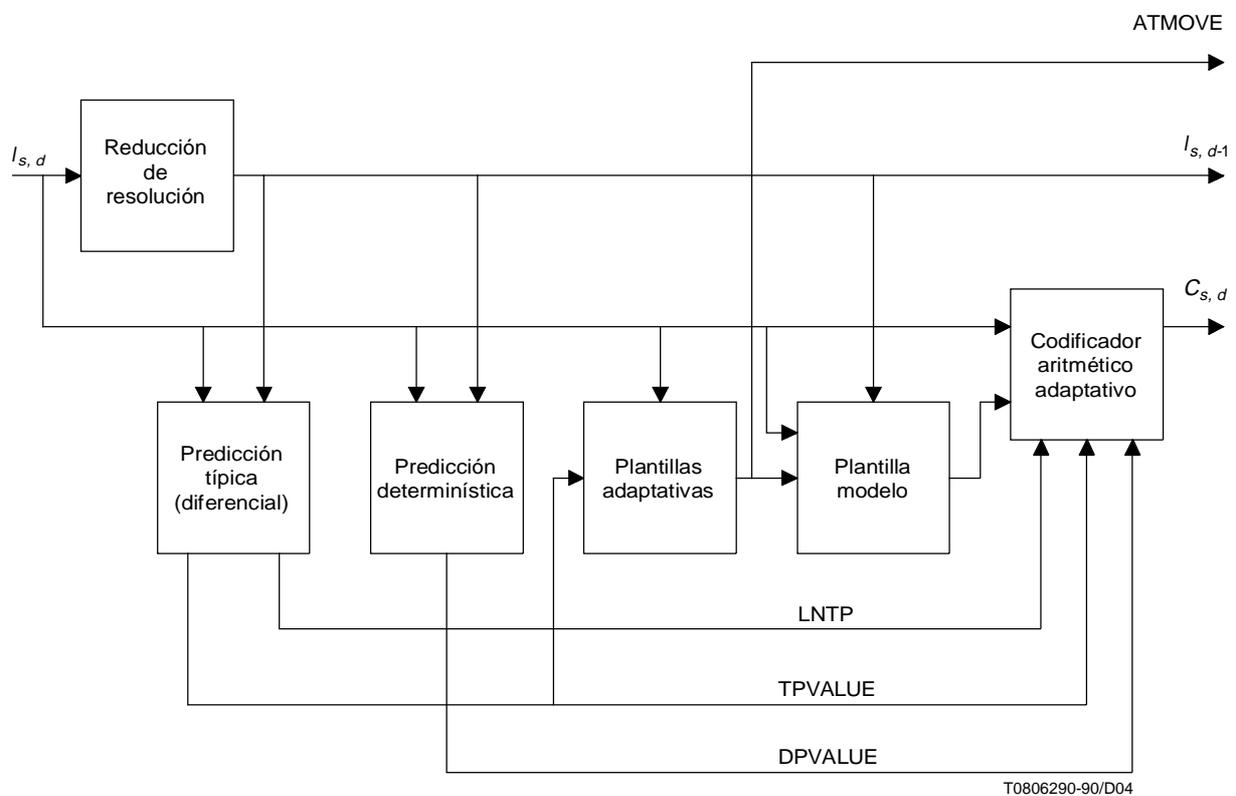


Figura Intro. 4 – Reducción de resolución y codificador de capa diferencial

Las siglas de los bloques de procesamiento de esta figura y de otras que se tratarán en esta introducción se indican en el Cuadro Intro. 3.

Cuadro Intro. 3 – Siglas de los bloques de procesamiento

Sigla	Significado
AAD	Decodificador aritmético adaptativo (<i>adaptive arithmetic decoder</i>)
AAE	Codificador aritmético adaptativo (<i>adaptive arithmetic encoder</i>)
AT	Plantillas adaptativas (<i>adaptive templates</i>)
DP	Predicción determinística (<i>deterministic prediction</i>)
MT	Plantillas modelo (<i>model templates</i>)
RR	Reducción de resolución (<i>resolution reduction</i>)
TPB	Predicción típica (inferior) [<i>typical prediction (bottom)</i>]
TPD	Predicción típica (diferencial) [<i>typical prediction (differential)</i>]

Intro. 3.1.1 Reducción de resolución

El bloque reducción de resolución (RR) efectúa la reducción de resolución. Este bloque acepta una imagen de alta resolución y crea una imagen de baja resolución que tiene, lo más aproximadamente posible, la mitad de filas y la mitad de columnas que el original.

Un modo evidente de dividir por dos la resolución de una determinada imagen en cada dimensión es submuestrearla tomando una de cada dos filas y una de cada dos columnas. El submuestreo es sencillo, pero crea imágenes de mala calidad subjetiva, especialmente cuando la imagen de entrada es binivel.

Con imágenes binivel que contienen texto y trazos de líneas, no es bueno utilizar submuestreo, pues suprime frecuentemente las líneas finas. Con imágenes binivel que contienen semitonos o tremolación ordenada para reproducir escalas de grises, tampoco lo es, ya que la grisicidad no está bien preservada, especialmente si el periodo de tremolación es una potencia de dos, como ocurre frecuentemente.

La presente Especificación sugiere un método de reducción de resolución. Este método concreto se ha diseñado cuidadosamente, comprobado extensamente, y se ha demostrado que consigue excelentes resultados con texto, dibujo lineal, escala de grises por tremolación, escala de grises por semitonos y escala de grises de error difuso.

Intro. 3.1.2 Predicción típica de capa diferencial

El bloque predicción típica (TP) de capa diferencial proporciona cierta ganancia de codificación, pero su objetivo primordial es acelerar las realizaciones. El TPD de capa diferencial busca regiones de color uniforme, y cuando comprueba que un determinado pixel vigente de alta resolución para codificación se halla en dicha región, no se necesita ninguno de los procesamientos normalmente efectuados en los bloques DP, AT, MT y AAE. En las imágenes de tipo texto o de dibujo lineal, el TP de capa diferencial permite usualmente evitar la codificación en el 95% de los pixels. En las imágenes binivel que reproducen la escala de grises, las economías de procesamiento son considerablemente menores.

Intro. 3.1.3 Predicción determinística

El objeto del bloque de predicción determinística (DP) es proporcionar ganancia de codificación. En el conjunto de imágenes de prueba utilizadas en el desarrollo de esta Especificación proporcionó una ganancia del 7%, que se considera típica.

Cuando se reduce la resolución de las imágenes con un determinado algoritmo de reducción de resolución, ocurre a veces que el valor de un determinado pixel vigente de alta resolución que ha de codificarse puede deducirse de los pixels ya conocidos por el codificador y el decodificador, esto es, de los pixels de la imagen de baja resolución y de los de la imagen de alta resolución causalmente relacionados (en un sentido raster) con el pixel vigente. Cuando así ocurre, se dice que el pixel vigente es determinísticamente predecible. El bloque DP indica cualquiera de esos pixels e inhibe su codificación por el codificador aritmético.

DP es un algoritmo de funcionamiento tabular. Los valores de los píxels circundantes de la imagen de baja resolución y de la imagen de alta resolución causal se utilizan para indexarlos en una tabla a fin de comprobar su determinicidad y, cuando ésta existe, obtener la predicción determinística. Las tablas DP son muy dependientes del método concreto de reducción de resolución utilizado. Se adoptan disposiciones para que un codificador telecargue tablas DP a un decodificador si está utilizando un algoritmo de reducción de resolución privado. Si una aplicación requiere DP por defecto, los decodificadores deben tener siempre las tablas DP por defecto, y no es necesario enviar ninguna tabla DP. Por tanto, si se utiliza el algoritmo de reducción de resolución sugerido, no es necesario enviar ninguna tabla DP.

Intro. 3.1.4 Plantillas modelo

Para cada píxel de alta resolución a codificar, el bloque plantillas modelo (MT) proporciona al codificador aritmético un entero denominado el contexto. Este entero viene determinado por los colores (niveles binarios) de los píxels concretos de la imagen de alta resolución causal, por los píxels concretos de la imagen de baja resolución ya disponible y por la fase espacial del píxel que se codifica. «Fase espacial» describe la orientación del píxel de alta resolución con relación a su correspondiente píxel de baja resolución.

El codificador aritmético mantiene para cada contexto una estimación de la probabilidad condicional del símbolo dado por ese contexto. La máxima ganancia de codificación se obtiene cuando esta estimación de probabilidad es al mismo tiempo exacta y próxima a 0 ó 1. Así, las buenas plantillas tienen un buen valor predictivo, de manera que cuando los valores de los píxels de la plantilla son conocidos, el valor del píxel a codificar es altamente predecible.

Intro. 3.1.5 Plantillas adaptativas

El bloque plantillas adaptativas (AT) proporciona una ganancia de codificación sustancial (a veces llega a un factor de 80%) en imágenes que reproducen la escala de grises por semitonos. AT busca una periodicidad horizontal en la imagen, y cuando la encuentra cambia la plantilla de manera que el píxel que precede al píxel vigente en exactamente con esta periodicidad se incorpore a la plantilla. Dicho píxel tiene un excelente valor predictivo.

Estos cambios son infrecuentes, y cuando se produce uno, se multiplexa una secuencia de control (indicada simbólicamente por *ATMOVE* en la Figura Intro. 4) para formar el tren de datos de salida. Por tanto, los decodificadores no necesitan efectuar ningún procesamiento para buscar la formación correcta de AT.

Intro. 3.1.6 Codificador aritmético adaptativo

El bloque codificador aritmético adaptativo es un codificador de entropía. Anota las salidas de los bloques TP y DP para determinar incluso si es necesario codificar un determinado píxel. Suponiendo que así es, anota entonces el contexto y utiliza su estimador de probabilidad interno para estimar la probabilidad condicional de que el píxel vigente sea de un color dado. A menudo el píxel es altamente predecible a partir del contexto, de manera que la probabilidad condicional es muy próxima a 0 ó 1, y puede tenerse una gran ganancia de codificación de entropía.

Mantener las estimaciones de probabilidad para cada uno de los contextos es un problema estadístico nada insignificante. Debe llegarse a un equilibrio entre la obtención de estimaciones sumamente exactas y la necesidad contrapuesta de adaptarse rápidamente a estadísticas subyacentes cambiantes.

Intro. 3.2 Codificador de capa de resolución mínima

La Figura Intro. 5 muestra un codificador de capa de resolución mínima. Es teóricamente más simple que el codificador de capa diferencial, dado que no son aplicables los bloques RR y DP. Para la definición de los nombres de las señales, refiérase a los cuadros de la cláusula 4. (No todos los subbloques deben ser utilizados en todos los sistemas.)

El TP de capa de resolución mínima, al igual que el TP de capa diferencial, está primordialmente destinado a acelerar el procesamiento. Los algoritmos utilizados para las dos versiones del TP son, no obstante, bastante diferentes, y no resulta posible saltar un porcentaje de píxels tan elevado con un TP de capa de resolución mínima con el TP de capa diferencial. En las imágenes que contienen texto y dibujo lineal, el TP de capa de mínima resolución permite el salto de aproximadamente el 40% de los píxels.

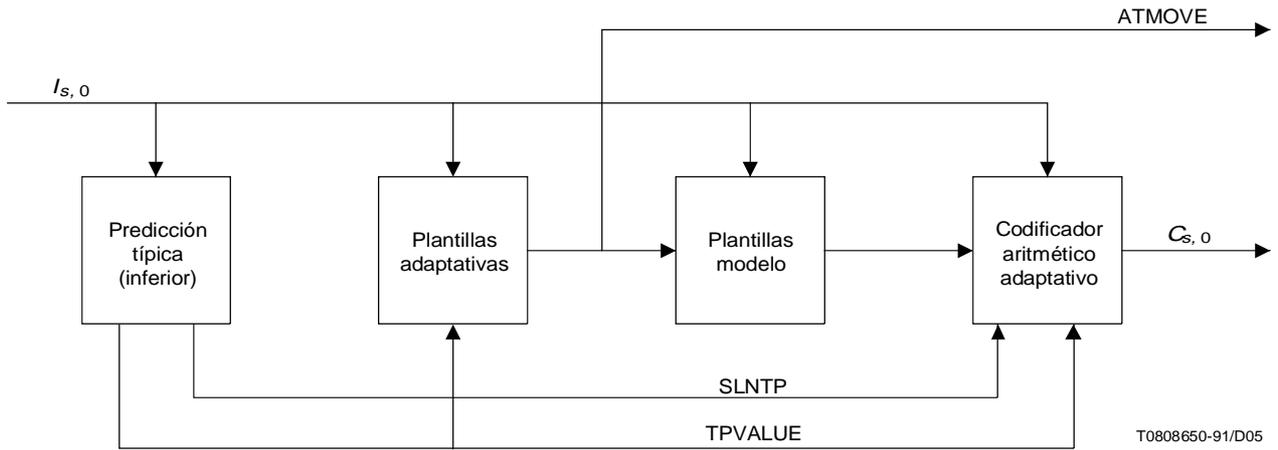


Figura Intro. 5 – Codificador de capa de resolución mínima

Intro. 4 Bloques funcionales del decodificador

Las Figuras Intro. 6, Intro. 7 e Intro. 8 son análogas a las Figuras Intro. 3, Intro. 4 e Intro. 5, pero presentan la decodificación y no la codificación. Obsérvese que los bloques RR y AT no aparecen en el decodificador. Para la definición de los nombres de las señales, refiérase a los cuadros de la cláusula 4. En codificación secuencial de una sola progresión, se utilizará el bloque decodificador de capa de resolución mínima de la Figura Intro. 6. No todos los bloques de las Figuras Intro. 7 e Intro. 8 tienen que ser utilizados en todos los sistemas.

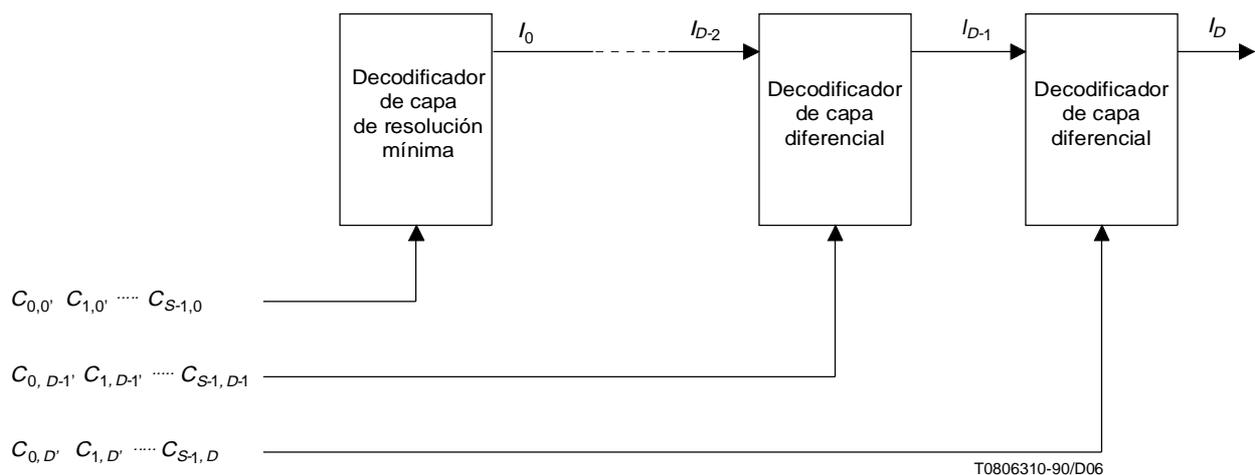


Figura Intro. 6 – Descomposición del decodificador

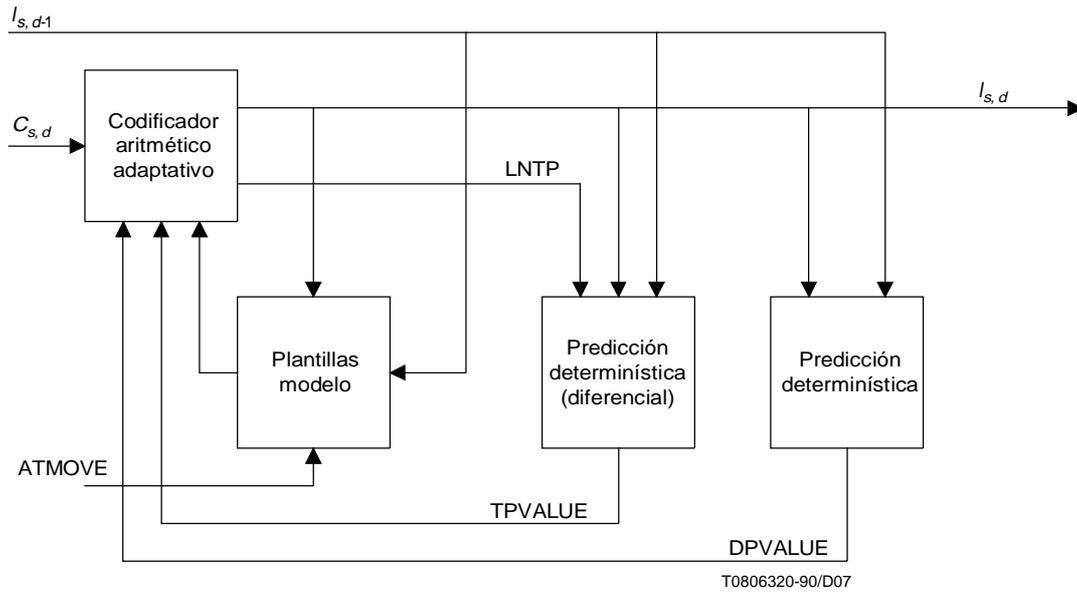


Figura Intro. 7 – Decodificador de capa diferencial

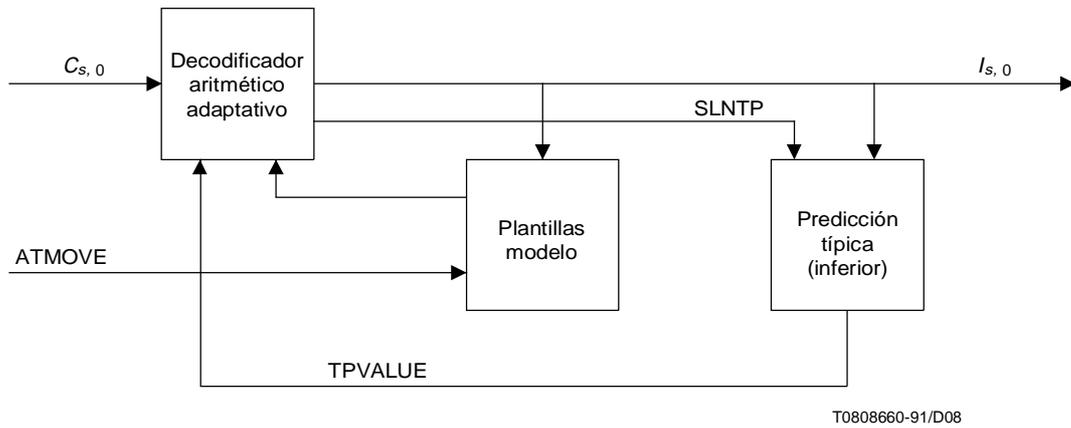


Figura Intro. 8 – Decodificador de capa de resolución mínima

NORMA INTERNACIONAL

RECOMENDACIÓN DEL CCITT

**TECNOLOGÍA DE LA INFORMACIÓN – REPRESENTACIÓN CODIFICADA
DE INFORMACIÓN DE IMAGEN Y DE AUDIO –
COMPRESIÓN DE IMAGEN BINIVEL PROGRESIVA****1 Alcance**

Esta Recomendación | Norma Internacional define un método de compresión preservador de bits (sin pérdida) para la codificación progresiva/secuencial compatible de planos de bits de imagen binivel, y es particularmente adecuada para imágenes binivel (de dos tonos, incluidas las de blanco y negro).

2 Referencias normativas

No hay referencias normativas. Las referencias informativas a normas y bibliografía técnica figuran en el Anexo F.

3 Definiciones

Para los fines de esta Recomendación | Norma Internacional, se aplican las siguientes definiciones.

- 3.1 codificador aritmético adaptativo:** Mecanismo para comprimir o descomprimir adaptativamente datos utilizando características de datos observadas para predecir y codificar futuros símbolos de datos.
- 3.2 plantillas adaptativas (AT):** Plantillas modelo que pueden modificarse desplazando un pixel AT durante el procesamiento de una imagen para sacar partido de los patrones observados en la imagen.
- 3.3 retardo AT:** Distancia en pixels entre el pixel que se codifica y el pixel AT.
- 3.4 pixel AT:** Pixel especial de la plantilla modelo al que se permite cambiar adaptativamente de posición durante el procesamiento de una imagen.
- 3.5 plano de bits:** Formación (o «plano») de símbolos binivel construidos a partir de una imagen eligiendo un determinado bit de cada pixel.
- 3.6 entrelazado de planos de bits:** Método utilizado para combinar dos o más planos de bits en un único tren de bits.
- 3.7 byte:** Ocho bits de datos.
- 3.8 relleno de bytes:** Mecanismo para distinguir inequívocamente entre bytes de escape predefinidos que indican el comienzo de un segmento marcador y bytes idénticos al byte de escape que se producen de manera natural en un tren de datos comprimido.
- 3.9 contexto:** Entero correspondiente al patrón específico de la plantilla y fase espacial (si se necesita) que se utiliza para identificar el índice del estado del codificador aritmético adaptativo que ha de utilizarse para codificar el pixel vigente.
- 3.10 predicción determinística (DP):** Método para predecir exactamente (y por tanto no codificar) determinados pixels de una imagen utilizando una versión de resolución más baja de la misma imagen con un conocimiento muy específico del método de reducción de resolución utilizado.
- 3.11 codificador de capa diferencial:** Mecanismo para codificar o decodificar imágenes de capa diferencial.
- 3.12 imagen de capa diferencial:** Imagen a una resolución dada que se describe especificando la diferencia entre ella misma y una imagen de menor resolución.
- 3.13 codificador de entropía:** Cualquier método sin pérdidas para comprimir datos.

- 3.14 byte de escape:** Byte de un tren de datos que significa que la información que sigue tiene un significado de código marcador especial.
- 3.15 pixel de alta resolución:** Pixel de la imagen de resolución superior de las dos capas de resolución en cuestión.
- 3.16 línea no típica (LNTF):** Condición que se produce durante la predicción típica cuando uno o más de los pixels asociados con una línea de baja resolución se predeciría incorrectamente.
- 3.17 codificador de capa de resolución mínima:** Mecanismo para codificar o decodificar imágenes de capa de resolución mínima.
- 3.18 imagen de capa de resolución mínima:** Imagen a una resolución dada que se describe sin referencia a cualesquiera imágenes de resolución inferior.
- 3.19 pixel de baja resolución:** Pixel de la imagen de resolución inferior de las dos capas de resolución en cuestión.
- 3.20 marcador:** Combinación de un byte de escape y un byte marcador que introduce información de control.
- 3.21 byte marcador:** Byte que sigue inmediatamente a un byte de escape que define el tipo de información de control que se introduce.
- 3.22 segmento marcador:** Combinación de un marcador y cualesquiera bytes adicionales de información de control asociada.
- 3.23 plantilla modelo (MT):** Patrón geométrico que describe la posición de los pixels con respecto a un pixel a codificar. Se utiliza para modelar características de imagen local.
- 3.24 pixel:** Elemento de una imagen que es descrita por una formación rectangular de tales elementos.
- 3.25 comportamiento progresivo:** Una técnica de codificación tiene comportamiento progresivo si una imagen es codificada primero como imagen de capa de resolución mínima y se aumenta luego sucesivamente la resolución mediante imágenes de capa diferencial.
- 3.26 codificación progresiva:** Método de codificación de imagen en el que ésta puede ser segmentada en barras, codificándose luego en primer lugar la imagen completa como imagen de capa de resolución mínima y aumentándose sucesivamente la resolución mediante imágenes de capa diferencial. Es compatible, mediante el reordenamiento de datos de barra/capa, con la codificación secuencial compatible progresiva.
- 3.27 codificación secuencial compatible progresiva:** Método de codificación de imagen en el que ésta puede ser segmentada en barras, que son codificadas secuencialmente; dentro de cada barra, la imagen es codificada progresivamente hasta la resolución completa. Es compatible, mediante el reordenamiento de datos de barra/capa, con la codificación progresiva.
- 3.28 datos codificados de barras protegidos (PSCD):** Tren de datos de imagen comprimidos que ha sido modificado mediante bytes de relleno para distinguir entre bytes de escape predefinidos que indican códigos marcadores especiales (que no forman parte del tren de datos comprimido) y bytes idénticos al byte de escape que se producen de manera natural en el tren de datos comprimidos.
- 3.29 método de reducción de resolución (RR):** Método para transformar una imagen de una determinada resolución en una imagen que describe el mismo objeto, pero con resolución menor.
- 3.30 comportamiento secuencial:** Una técnica de codificación tiene comportamiento secuencial si las porciones de la imagen cercanas a la parte superior quedan completamente descritas antes de que las porciones inferiores empiecen a ser descritas.
- 3.31 codificación secuencial de una sola progresión:** Método de codificación de imagen en el que ésta es totalmente codificada en una sola capa de resolución, línea por línea, de izquierda a derecha y de arriba abajo, sin referencia a ninguna imagen de resolución inferior. Es compatible con la codificación progresiva y con la codificación secuencial compatible progresiva si el número de capas diferenciales es cero.
- 3.32 fase espacial:** Atributo de un pixel en una imagen de capa diferencial que describe su orientación con respecto al pixel de baja resolución asociado con el mismo.
- 3.33 resolución espacial:** Número de pixels utilizados para describir una región de una imagen de tamaño espacial fijo.

3.34 barra: Región de tamaño vertical fijo de una imagen que comprende toda la anchura horizontal de dicha imagen.

3.35 pixel objetivo: Pixel que ha de ser procesado.

3.36 predicción típica (TP): Método para predecir exactamente (y por tanto no codificar) bloques de pixels de una imagen explotando grandes regiones de color uniforme.

4 Símbolos y siglas

4.1 Siglas

Véase el Cuadro 1.

Cuadro 1 – Siglas

Siglas	Significado
AT	Plantillas adaptativa (<i>adaptive templates</i>)
DP	Predicción determinística (<i>deterministic prediction</i>)
LPS	Símbolo menos probable (<i>less probable symbol</i>)
LSB	Bit menos significativo (<i>least significant bit</i>)
MPS	Símbolo más probable (<i>more probable symbol</i>)
MSB	Bit más significativo (<i>most significant bit</i>)
MT	Plantillas modelo (<i>model templates</i>)
RR	Reducción de resolución (<i>resolution reduction</i>)
TP	Predicción típica (<i>typical prediction</i>)

4.2 Constantes simbólicas

Véase el Cuadro 2.

Cuadro 2 – Símbolos de constantes

Constante	Significado	Valor	
		ISO	Hexadecimal
ABORT	Aborto	00/04	0x04
ATMOVE	Desplazamiento de AT	00/06	0x06
COMMENT	Comentario privado	00/07	0x07
ESC	Escape	15/15	0xff
NEWLEN	Nueva longitud	00/05	0x05
RESERVE	Reserva	00/01	0x01
SDNORM	Fin de datos de barra normal	00/02	0x02
SDRST	Reiniciación en fin de datos de barra	00/03	0x03
STUFF	Relleno	00/00	0x00

4.3 Símbolos matemáticos, operadores e indicadores

Véase el Cuadro 3.

4.4 Variables con nombres mnemotécnicos

Véase el Cuadro 4.

5 Convenios

5.1 Convenios y símbolos de los diagramas de flujo

La entrada a todos los diagramas de flujo se produce por la parte superior y la salida por la inferior. El símbolo «<<<» designa un desplazamiento izquierdo primario con relleno nulo de bits de bajo orden y el símbolo «>>>» designa un desplazamiento derecho binario con relleno nulo de bits de alto orden. Tanto para «<<<» como para «>>>», la magnitud de la izquierda es la que se desplaza y la magnitud de la derecha es la cantidad de desplazamiento. El AND lógico binario de dos números se indica por «&».

5.2 Gráficos de plantilla

Frecuentemente será necesario mostrar gráficamente la relación de los pixels de una imagen de alta resolución a los pixels de una imagen de baja resolución. La Figura 1 es un gráfico tridimensional que muestra dicha relación. En el texto de esta Especificación se utilizan en su lugar representaciones bidimensionales como las de la Figura 2, por ser más compactas que su equivalente tridimensional. Nótese que los pixels de baja resolución se representan por círculos y que los correspondientes pixels de alta resolución de gráficos bidimensionales se representan por cuadrados, parcialmente ocultos por los círculos de baja resolución.

Cuadro 3 – Símbolos matemáticos, operadores e indicadores

Símbolo	Significado
$C_{s, d}$	Datos codificados de la barra s y la capa d
D	Número para la capa de resolución final en la BIE
D_L	Número para la capa de resolución inicial en la BIE
h	Pixel de alta resolución
I_d	Imagen en la capa d
l	Pixel de baja resolución
L_c	Longitud en bytes de comentario privado
L_d	Líneas por barra en la capa d
M_X	Máximo desplazamiento horizontal permitido en el pixel AT
M_Y	Máximo desplazamiento vertical permitido en el pixel AT
p	Probabilidad
P	Número de planos de bits
R_d	Resolución en la capa d
S	Número de barras
X_d	Tamaño de imagen horizontal en la capa d
y_{AT}	Línea en la que ha de hacerse una conmutación AT
Y_d	Tamaño de imagen vertical en la capa d
τ_X	Desplazamiento horizontal del pixel AT
τ_Y	Desplazamiento vertical del pixel AT
$>>$	Desplazamiento binario hacia la derecha
$<<$	Desplazamiento binario hacia la izquierda
$\&$	AND lógico
$!$	NOT lógico
\oplus	OR exclusivo
$\lceil \cdot \rceil$	Función tope (entero más pequeño \geq argumento)
0x	Indicador hexadecimal

5.3 Fase espacial

Puede asociarse una «fase espacial» con los pixels en cualquier capa de resolución distinta de la mínima. Esta «fase» describe la orientación del pixel con respecto al pixel de resolución inferior asociado con el mismo. Si se trata del pixel superior izquierdo de los cuatro pixels asociados con un pixel de baja resolución, se dirá que tiene «fase 0». Análogamente, el pixel superior izquierdo tendrá «fase 1», el pixel inferior izquierdo tendrá «fase 2», y el pixel superior derecho tendrá la «fase 3» (véase la Figura 3).

Cuadro 4 – Variables con nombres mnemotécnicos

Variable	Significado
A	Registro de tamaño de intervalo
BID	Datos de imagen binivel
BIE	Entidad de imagen binivel
BIH	Encabezador de imagen binivel
BUFFER	Memoria intermedia
C	Registro de código
CE	Intercambio condicional
CHIGH	Registro de código, dos bytes altos
CLOW	Registro de código, dos bytes bajos
CT	Contador de bits
CX	Contexto
DPLAST	Ultima DP
DPON	DP habilitada
DPPRIV	DP privada
DPTABLE	Tabla de DP
DPVALUE	Valor de DP
HITOLO	De alto a bajo
I LEAVE	Intercalación de múltiples planos de bits
LNTP	Línea no típica
LPIX	Pixel de baja resolución
LRLTWO	Plantilla de dos líneas de capa de mínima resolución
LSZ	Tamaño de LPS en el intervalo de codificación
MPS	Símbolo más probable
NLPS	Siguiente si LPS
NMPS	Siguiente si MPS
PIX	Pixel
PSCD	Datos codificados de barra protegida
SC	Contador de pilas (de bytes 0xff)
SCD	Datos codificados de barra
SDE	Entidad de datos de barra
SEQ	Secuencial
SLNTP	Misma LNTP
SMID	Indice sobre barra está en medio
SWTCH	Conmutación
ST	Estado
TPBON	TP de capa de mínima resolución habilitada
TPDON	TP de capa diferencial habilitada
TPVALUE	Valor de TP
VLENGTH	Longitud variable

5.4 Gráficos de estructura de datos

Los Cuadros 5 a 16 contienen gráficos que ilustran la descomposición de campos en subcampos. Tipográficamente se transmitirán primero los subcampos situados más a la izquierda. La última fila de cada uno de estos gráficos da los tamaños de campo en bytes. Una entrada «1/8» indica un solo bit. Se utiliza una entrada «Varía» cuando el tamaño del campo es variable y depende de las opciones elegidas, los parámetros elegidos, o de la imagen concreta que se codifica.

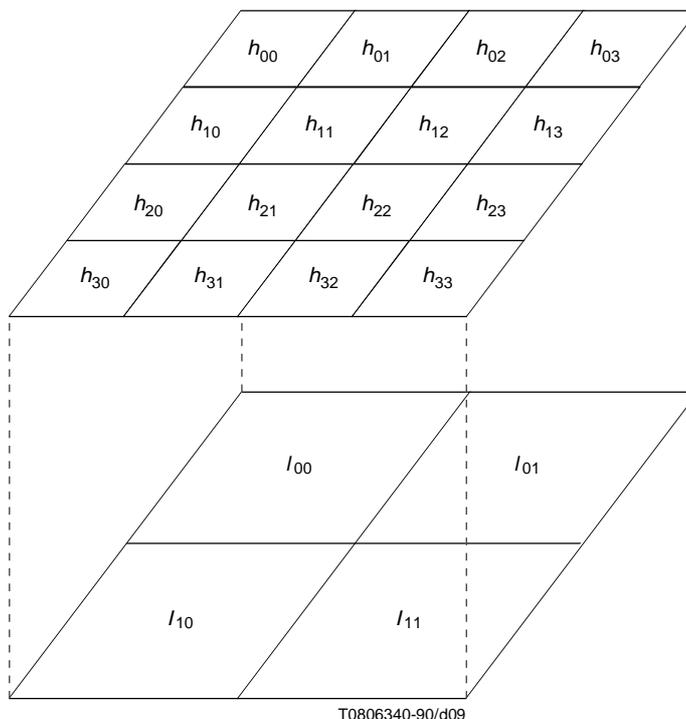


Figura 1 – Pixels de alta y baja resolución en gráfico tridimensional

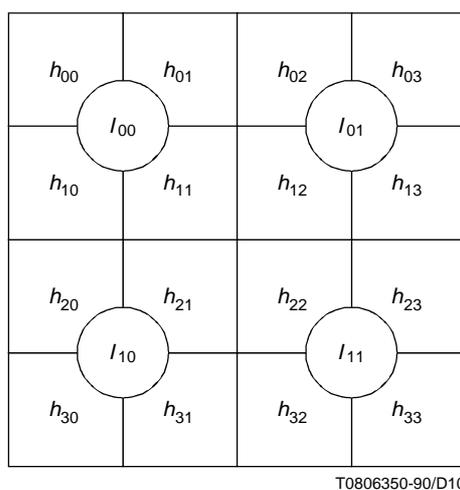


Figura 2 – Pixels de alta y baja resolución en gráfico bidimensional

6 Requisitos

6.1 Reglas generales

6.1.1 Asignación de color

Cada bit de cada plano de pixels es 0 ó 1. Cuando la imagen es binivel, un bit 1 indicará el color de primer plano y un bit 0 indicará el color de fondo. Si existe más de un plano de bits, la correspondencia de la intensidad y el color con planos de bits no es definido por esta Especificación.

NOTA – Si 1 ó 0 representa el color de primer plano, es irrelevante para todos los aspectos de esta Especificación, excepto el método de reducción de resolución descrito. Este método de reducción de resolución tiene una ligera asimetría entre el blanco y el negro.

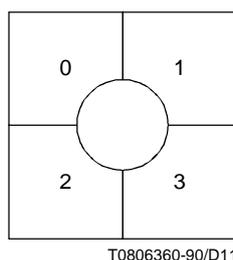


Figura 3 – Cuatro posibles fases de los pixels de alta resolución

6.1.2 Convenios de bordes

La reducción de resolución, la predicción típica, la predicción determinística y los algoritmos de codificación iterarán todos a través de la imagen en el orden de exploración raster usual, es decir, de izquierda a derecha y de arriba a abajo. El procesamiento de un pixel objetivo vigente referenciará los colores de algunos pixels en una relación espacial fija con respecto a ese pixel objetivo. En los bordes de la imagen, estas referencias vecinas pueden no hallarse en la imagen real. Las reglas para satisfacer las referencias fuera de imagen serán las siguientes:

- se supondrá un borde blanco (0) en la parte superior, izquierda y derecha, de la imagen real;
- el fondo de la imagen se extenderá hacia abajo en la medida necesaria por reproducción con pixels de la última línea efectiva de la imagen.

Además, al referenciar los pixels a través de fronteras de barras, se aplicarán las siguientes reglas:

- una referencia de pixel en una barra por encima de la vigente retornará el valor real del pixel, a menos que el pixel esté por encima de la imagen, en cuyo caso se aplicará la regla del borde blanco a la parte superior de la imagen;
- una referencia de pixel en una barra por debajo de la vigente será satisfecha por reproducción con pixels de la última línea de la barra vigente. En particular, no se utilizarán valores efectivos ni siquiera si la referencia está dentro de la imagen.

NOTA – Esta última regla sólo es pertinente para la imagen de baja resolución, ya que para la decodificabilidad nunca puede haber referencias a los pixels de alta resolución de la fila inferior. Además, ocurre que el algoritmo de reducción de resolución descrito nunca referencia ni siquiera los pixels de baja resolución de la línea situada más abajo.

6.1.3 Alineación de bytes

NOTA – Dados los convenios de encabezamiento y de código marcador descritos en 6.2, las secuencias de escape están siempre alineadas dentro de los trenes de datos.

6.2 Organización de los datos

6.2.1 Descomposición de la imagen

La estructura de datos del máximo nivel descrita en esta Especificación se conoce como entidad de imagen binivel (BIE, *bi-level image entity*). Una BIE dada puede contener datos para una o más capas de resolución y planos de bits. Los datos que describen una imagen dada en todas sus resoluciones y planos de bits disponibles pueden estar contenidos, aunque no necesariamente, en más de una BIE.

NOTA – Se necesita una descripción de múltiples BIE de una imagen cuando las imágenes se ponen primero disponibles a una resolución o una precisión de plano de bits baja o intermedia, y puede o no haber una petición de potenciación a una resolución o precisión superior.

6.2.2 Descomposición de la entidad de imagen binivel y del encabezamiento (BIE y BIH)

Como se muestra en el Cuadro 5, una entidad de imagen binivel (BIE) comprenderá un encabezamiento de imagen binivel (BIH, *bi-level image header*) y datos de imagen binivel (BID, *bi-level image data*).

Cuadro 5 – Descomposición de BIE

BIE	
BIH	BID
Varía	Varía

El encabezamiento de imagen binivel comprenderá los campos mostrados en los Cuadros 6, 7 y 8.

Cuadro 6 – Descomposición de BIH

BIH											
D_L	D	P	–	X_D	Y_D	L_0	M_X	M_Y	Orden	Opciones	DPTABLE
1	1	1	1	4	4	4	1	1	1	1	0 ó 1728

Cuadro 7 – Descomposición del byte orden

Orden							
MSB	...						LSB
–	–	–	–	HITOLO	SEQ	I LEAVE	SMID
1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8

Cuadro 8 – Descomposición del byte opciones

Opciones							
MSB	...						LSB
–	LRLTWO	VLENGTH	TPDON	TPBON	DPON	DPPRIV	DPLAST
1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8

El primer byte de la BIH especificará D_L , la capa de resolución inicial que ha de especificarse en esta BIE. Con mucha frecuencia, este número será cero, en cuyo caso los datos transmitidos permitirán componer la imagen sin ningún conocimiento previo de la misma. Será distinto de cero una BIE ya ha definido la imagen a una capa intermedia, y sólo debe especificarse información incremental. El segundo byte especifica D , la capa de resolución final descrita en esta BIE. Obsérvese que con tales BIE múltiples, cuando D_L es nula, D es igual a un número de capas diferenciales, pero no al número total de capas diferenciales.

El tercer byte especificará P , número de planos de bits. Si la imagen es binivel, P será 1.

El cuarto byte es de relleno. Se escribirá siempre 0.

Los tres siguientes campos de cuatro bytes especifican X_D , Y_D y L_0 que son, respectivamente, la dimensión horizontal a la resolución máxima, la dimensión vertical a la resolución máxima, y el número de bytes por barra a la resolución mínima. Estos tres enteros se codifican empezando por el byte más significativo. En otras palabras, X_D es la suma de 256^3 veces el quinto byte de BIH, 256^2 veces el sexto byte, 256 veces el séptimo byte, y el octavo byte.

El decimoséptimo y decimoctavo bytes especificarán M_X y M_Y , los máximos desplazamientos horizontales y verticales permitidos para el pixel AT. Estos parámetros se tratan más detenidamente en 6.7.3.

El decimonoveno byte de la BIH transportará los parámetros binarios HITOLO, SEQ, ILEAVE y SMID, que juntos especifican el orden en el que los datos de barra se concatenan para formar BID. En 6.2.4 se dan más detalles. Los cuatro bits más significativos de este byte son de relleno y se escribirán siempre 0.

El duodécimo byte del BIH especificará las opciones. Su bit más significativo es de relleno y se escribirá siempre 0. La plantilla utilizada para codificar la capa de mínima resolución tendrá 2 ó 3 líneas ya que LRLTWO es 1 ó 0 (véase 6.7.1). Si el bit VLENGTH es 0, no habrá códigos marcadores NEWLEN (véase 6.2.6.2). Si VLENGTH es 1, puede haber o no presentes códigos marcadores NEWLEN. Los bits TPDON, TPBON y DPON son 1 cuando se desea habilitar respectivamente TP de capa diferencial, TP de capa de resolución mínima y DP. Los bits DPPRIV y DPLAST son sólo significativos si DPON es igual a 1. Si DPON es 1 y DPPRIV es 1, debe utilizarse una tabla DP privada. Si DPLAST es 0, ha de cargarse la tabla DP privada. En otro caso, debe reutilizarse la última tabla DP utilizada.

El campo DPTABLE de BIH sólo estará presente si DPON es igual a 1, DPPRIV igual a 1, y DPLAST igual a 0. Su tamaño e interpretación se definen en 6.6.

Las variables D_L , D , P , X_D , Y_D , L_0 , M_X , M_Y , HITOLO, SEQ, ILEAVE, SMID, LRLTWO, VLENGTH, TPDON, TPBON, DPON, DPPRIV y DPLAST son parámetros libres. Algunas normas de aplicaciones pueden restringir la selección de algunas o todas ellas. El Cuadro 9 muestra los límites de estos parámetros tal como vienen determinados por sus naturalezas implícitas o los tamaños de campo admitidos para ellos en BIH.

Un tren de datos JBIG es cualquier tren de datos creado como se describe en las partes normativas de esta Especificación, con parámetros comprendidos dentro de la gama del Cuadro 9. Con miras a crear una comunidad lo más grande posible de aplicaciones a las que les resulte posible compartir soporte físico e intercambiar datos decodificables, el Anexo A sugiere sustentación mínima de estos parámetros libres. Se alienta a las diferentes aplicaciones a escoger valores de parámetros comprendidos en las gamas sugeridas de sustentación mínima siempre que sea posible hacerlo. Para las realizaciones que deseen ser compatibles con una amplia gama de aplicaciones puede ser conveniente sustentar todas las elecciones de parámetros libres dentro de las gamas sugeridas.

6.2.3 Iteración con parámetros dependientes de la capa de resolución

Las dimensiones de la imagen en las capas inferiores (indizadas por d) se definirán recurrentemente para $D \geq d \geq 1$ por las iteraciones

$$X_{d-1} = \lceil X_d / 2 \rceil \quad (1)$$

$$Y_{d-1} = \lceil Y_d / 2 \rceil \quad (2)$$

donde $\lceil \cdot \rceil$ designa la función tope o, en otras palabras, el menor entero mayor o igual que el argumento.

Cuadro 9 – Límites absolutos de los parámetros libres

Parámetro	Mínimo	Máximo
D_L	0	D
D	D_L	255
P	1	255
X_D	1	4 294 967 295
Y_D	1	4 294 967 295
L_0	1	4 294 967 295
M_X	0	127
M_Y	0	255
HITOLO	0	1
SEQ	0	1
ILEAVE	0	1
SMID	0	1
LRLTWO	0	1
VLENGTH	0	1
TPDON	0	1
TPBON	0	1
DPON	0	1
DPPRIV	0	1
DPLAST	0	1

Para $1 \leq d \leq D$, el número de líneas por barra en la capa D se definirá por

$$L_d = 2 \times L_{d-1} \tag{3}$$

En todas las capas habrá necesariamente

$$S = \lceil Y_0 / L_0 \rceil \tag{4}$$

barras. En muchos casos, la última barra de cualquier capa d tendrá menos de L_d líneas.

6.2.4 Descomposición de los datos de imagen binivel (BID)

Los datos codificados $C_{s,d,p}$ que definen una determinada barra s a una resolución d y un plano de bits p estarán contenidos en una entidad de datos de barra o SDE. La BID constará de una concatenación de SDE y los códigos marcadores flotantes mostrados en el Cuadro 10.

Cuadro 10 – Descomposición de BID

BID						
Código(s) marcador(es) flotante(s)	SDE _{s, d, p}	Código(s) marcador(es) flotante(s)	SDE _{s, d, p}	...	Código(s) marcador(es) flotante(s)	SDE _{s, d, p}
Varía	Varía	Varía	Varía	...	Varía	Varía

La ordenación de los SDE depende de HITOLO, SEQ, ILEAVE y SMID. La jerarquización de índices será la definida por el Cuadro 11. Se admiten únicamente las seis combinaciones de las tres variables SEQ, ILEAVE y SMID indicadas en el Cuadro 11. Las dos combinaciones restantes no ocurrirán nunca. Los bucles sobre las variables ficticias s y p van de 0 a $S-1$ (de arriba a abajo) y de $P-1$ a 0 (del MSB al LSB). Si HITOLO es 0, la variable d oscilará de D_L a D . En otro caso, oscilará de D a D_L .

En el Cuadro Intro. 2 se presenta un ejemplo didáctico.

Cuadro 11 – Ordenación de codificaciones de barra en BID

			Bucles		
SEQ	ILEAVE	SMID	Exterior	Medio	Interior
0	0	0	p	d	s
0	1	0	d	p	s
0	1	1	d	s	p
1	0	0	s	p	d
1	0	1	p	s	d
1	1	0	s	d	p

6.2.5 Descomposición de la entidad de datos de barra (SDE)

Como muestra el Cuadro 12, cada SDE estará terminada por un byte ESC y un byte SDNORM o un byte SDRST.

Cuadro 12 – Estructura de la SDE

SDE		
PSCD	ESC	SDNORM o SDRST
Varía	1	1

Normalmente el byte de terminación será SDNORM. Si en cambio el byte de terminación es SDRST, el «estado» para ese plano de bits determinado y esa capa de resolución determinada será reiniciado antes de la codificación o decodificación de la próxima barra de ese plano y capa. La reposición del estado exige la inicialización de los estimadores de probabilidad adaptativos como en la parte superior de la imagen, la reposición (si es necesario) del pixel AT en ubicación por defecto, y la inicialización de $LNTp_{y-1}$ a 1 cuando se está en la capa de resolución mínima. También es necesario que todas las funciones que incluyen reducción de resolución, predicción determinística, predicción típica y plantillas modelo inicien la barra siguiente como lo hacen en la parte superior de la imagen, es decir, como se define en 6.1.2.

NOTA – La reposición del estado con SDRST no debe hacerse innecesariamente, ya que al hacerlo se degrada la eficacia de compresión, lo que puede introducir perturbaciones en los límites de las barras en las imágenes de menor resolución.

Los datos codificados de barra protegida (PSCD) se definen como los bytes de la SDE que quedan después de eliminar los dos bytes de terminación. Un decodificador creará datos codificados de barra (SCD) a partir de los PSCD sustituyendo todas las apariciones de un byte ESC seguido por un byte STUFF con un único byte ESC. Un codificador creará PSCD a partir de SCD sustituyendo todas las apariciones de un byte ESC por un byte ESC seguido por un byte STUFF. La utilización de SCD se describe en 6.8. PSCD se utiliza al definir la SDE más que SCD para que los datos de una barra puedan situarse inequívocamente en los BID.

Pueden utilizarse un byte ESC y un byte ABORT para terminar prematuramente los BID, como se muestra en el Cuadro 13.

NOTA – Sin un mecanismo como éste un codificador que encuentra un problema dejaría indefinidamente «colgado» a su decodificador asociado. No habría modo de rearmar el codificador, ya que no se repondría hasta después de que hubiera recibido el volumen de datos anunciado.

Cuadro 13 – Código marcador para terminar prematuramente los BID

ESC	ABORT
1	1

6.2.6 Segmentos marcadores flotantes

Los segmentos marcadores flotantes proporcionan información de control. No aparecerán dentro de una SDE. Pueden aparecer entre SDE o antes de la primera. Hay tres segmentos marcadores flotantes: ATMOVE, NEWLEN y COMMENT. (Véase la Nota 2 de 6.2.6.2.)

6.2.6.1 Desplazamiento de la plantilla adaptativa (AT)

La posición del pixel AT puede cambiarse con el segmento marcador ATMOVE que se muestra en el Cuadro 14.

Cuadro 14 – Estructura del segmento marcador flotante ATMOVE

ESC	ATMOVE	y_{AT}	τ_X	τ_Y
1	1	4	1	1

Los bytes tres, cuatro, cinco y seis definen y_{AT} , línea en la que cambia la plantilla. Los bytes siete y ocho definen τ_X y τ_Y , los desplazamientos horizontal y vertical del nuevo pixel AT. La línea y_{AT} se decodificará como la suma de 256^3 veces el tercer byte, 256^2 veces el cuarto byte, 256 veces el quinto byte y el sexto byte.

El estimador de probabilidad no se reinicializa después de un ATMOVE. La capa de resolución y el plano de bits para el cual ha de ser efectivo un determinado segmento marcador ATMOVE será el de la primera SDE que sigue al segmento marcador. La numeración de línea de y_{AT} comienza en 0 para cada barra para que si, por ejemplo, un cambio ha de ser efectivo en la línea inicial de una barra, y_{AT} sea igual a 0.

En 6.7.3 se tratan más detenidamente los pixels AT y las variables y_{AT} , τ_X y τ_Y .

6.2.6.2 Redefinición de la longitud de imagen

Si VLENGTH es 1, es admisible cambiar la longitud de Y_D de la imagen con un código marcador de nueva longitud, como muestra el Cuadro 15.

Cuadro 15 – Segmento marcador para indicar una nueva dimensión vertical

ESC	NEWLEN	Y_D
1	1	4

Como máximo sólo aparecerá un segmento marcador de nueva longitud en cualquier BIE. Ahora bien, un segmento marcador podría hacer referencia a una línea situada en la barra precedente inmediata debido a una terminación inesperada de la imagen o a la utilización de una sola barra. Dicho segmento marcador va seguido inmediatamente de ESC + SDNORM/SDRST, y el nuevo Y_D dado por el segmento marcador de nueva longitud puede ser inferior al número de línea del final de la barra precedente. El codificador no codificará más que el número de líneas de cada capa correspondiente al nuevo Y_D . Dentro del segmento marcador de nueva longitud, Y_D se empaquetará en su cuarto campo de byte exactamente como se empaqueta en su cuarto campo de byte en BIH. El nuevo Y_D nunca será mayor que el original.

NOTAS

1 El marcador de nueva longitud se ha definido de forma que sea posible comenzar la codificación de una imagen de longitud desconocida. En este caso, la dimensión original Y_D colocada en el encabezamiento sirve de indicación de la máxima longitud que podría tener la imagen.

2 Algunas aplicaciones pudieran requerir que el nuevo segmento marcador de longitud siguiese inmediatamente a un PSCD justo antes de ESC + SDNORM/SDRST para terminar prematuramente. Esta utilización se encuentra en estudio.

6.2.6.3 Segmento marcador de comentario

Un segmento marcador de comentario empezará por un byte ESC seguido de un byte COMMENT y un entero L_c de cuatro bytes, como se muestra en el Cuadro 16.

El número L_c será igual a la suma de 256^3 veces el tercer byte, 256^2 veces el cuarto byte, 256 veces el quinto byte, y el sexto byte. Este número dará solamente la longitud de la porción comentario privado del segmento marcador de comentario. Es decir, la longitud total del segmento marcador de comentario será $L_c + 6$ bytes.

Cuadro 16 – Segmento marcador de comentario

ESC	COMMENT	L_c
1	1	4

6.2.7 Byte marcador reservado

Un byte ESC seguido por un byte RESERVE es un marcador reservado. En 6.8.2.8 se describe un posible uso de este marcador. Las futuras extensiones de esta Especificación no utilizarán este marcador para ningún fin. Los codificadores o decodificadores pueden emplearlo para cualquier fin privado deseado. El marcador reservado nunca aparecerá en un tren de datos público.

6.3 Reducción de resolución

La tabla de predicción determinística por defecto es adaptada al algoritmo de reducción de resolución descrito en esta subcláusula. Es admisible utilizar un algoritmo de reducción de resolución alternativo, pero en ese caso debe neutralizarse la predicción determinística o bien telecargarse al decodificador una tabla de predicción determinística adaptada como parte del BIH.

El algoritmo de reducción de resolución es idéntico para todas las capas de resolución y todos los planos de bits. El procesamiento para crear la imagen en la capa de resolución $d - 1$ a partir de una imagen en la capa de resolución d se describe a continuación.

Si X_d o Y_d no es par, para los fines de reducción de resolución se creará una nueva imagen de capa d añadiendo, si es necesario, una columna blanca a la derecha o una reproducción de pixels de la última línea en el fondo. Por tanto, en el resto de este punto X_d e Y_d se suponen pares.

La imagen original puede dividirse en dos por dos bloques de pixels, y cada uno de estos dos por dos superpixels se corresponderá con un pixel de baja resolución en la imagen de resolución reducida. Estos pixels de baja resolución se determinarán de izquierda a derecha y de arriba a abajo en el orden de exploración raster normal. La regla de reducción de esta Especificación será la definida en la Figura 4 y el Cuadro 17. El razonamiento que fundamenta este mapeamiento concreto se explica en el Anexo B (informativo).

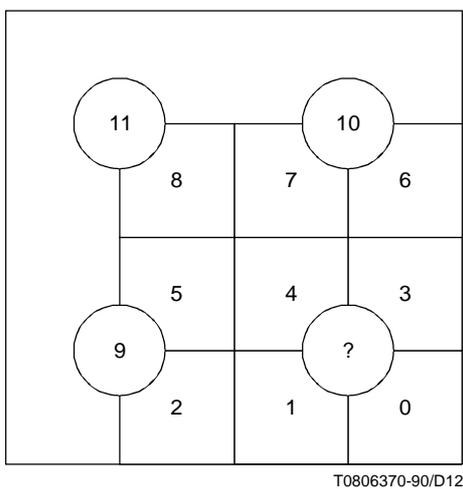


Figura 4 – Píxeles utilizados para determinar el color de un píxel de baja resolución

El círculo que tiene «?» en su interior representa el píxel de baja resolución cuyo color debe determinarse. Los círculos y cuadrados con números en su interior corresponden a los píxeles utilizados para hacer esta determinación.

Los colores de los píxeles numerados definen un índice, definiendo cada píxel numerado un bit del índice. El píxel designado por «0» determina el bit menos significativo del índice, y cada píxel numerado adicional determina el bit del índice correspondiente a su número. Cuando un píxel toma el color del primer plano, su bit correspondiente del índice tomará el valor «1». Dado este índice, el color del píxel designado por «?» será el definido por el Cuadro 17, que está indizado de izquierda a derecha. Por ejemplo, los colores de los píxeles correspondientes a los índices 0 a 7 son respectivamente 0, 0, 0, 1, 0, 0, 0, 1.

En los bordes de una imagen, algunos de los píxeles numerados de la Figura 4 pueden no estar dentro de la imagen. Para la definición de un índice, se utilizarán las reglas generales aplicables a los bordes indicadas en 6.1.2.

Al empezar la reducción de resolución, el píxel situado en el extremo superior izquierdo de la imagen de alta resolución será alineado con el píxel 4 de la Figura 4.

6.4 Predicción típica de capa diferencial

La TP de capa diferencial será activada o neutralizada con el bit TPDON en el campo opciones de BIH. Si se neutraliza (TPDON = 0), el bloque TPD de un codificador o de un decodificador producirá simplemente TPVALUE = 2 para todos los píxeles, para indicar al bloque de codificación o decodificación aritmética que no se está haciendo predicción. Además, cuando se neutraliza TP de capa diferencial, el seudopíxel LNTP no será codificado ni decodificado por el codificador aritmético. En el resto de este punto y sus subpuntos se supone que TP de capa diferencial está activada (TPDON = 1).

Siempre que se haga referencia a un píxel que debido a los efectos de bordes no esté realmente en la barra vigente, el valor de este píxel será determinado por las reglas de bordes generales de 6.1.2.

6.4.1 Procesamiento de un codificador

La Figura 5 define una vecindad 8. Los ocho píxeles que no tienen «?» son inmediatamente adyacentes al mismo, y son su vecindad 8.

Un píxel de baja resolución dado es «no típico» si él y todos los demás píxeles de su vecindad 8 son del mismo color, pero uno o más de los cuatro píxeles de alta resolución asociados con el mismo difiere(n) de este color común. Una línea dada de baja resolución es «no típica» (LNTP) si contiene píxeles «no típicos». En la Figura 6 se muestra un diagrama de flujo para procesamiento a fin de determinar LNTP. En esta figura LPIX designa un píxel de baja resolución.

NOTA – Los píxeles de baja resolución que no son típicos en este sentido son posibles, pero muy poco frecuentes.

Cuadro 17 – Mapa de determinación del color de baja resolución

Índice	Color							
[0, 63]	00010001	01110011	11111111	11111111	00110011	11111111	11111111	11111111
[64, 127]	00000001	01110111	11111111	11111111	00110111	11111111	11111111	11111111
[128, 191]	00110111	11111111	11111111	11111111	01111101	11111111	11111111	11111111
[192, 255]	00110111	11111111	11111111	11111111	11111111	01111101	11111111	11111111
[256, 319]	00000001	00110111	11111101	11111111	00111111	11111111	11111111	11111111
[320, 383]	00110111	01111111	11111111	01111111	01111111	01111111	01111111	11111111
[384, 447]	00110101	11111111	11110111	11111111	11011111	01111111	11111111	11111111
[448, 511]	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
[512, 575]	00000001	00100011	00000101	00111011	00010001	00100011	01110001	11111111
[576, 639]	00000001	01110101	00111011	01111111	00000000	01010011	11111110	11111111
[640, 703]	00000001	01000001	01111111	11111111	00001001	10110111	11111111	11111111
[704, 767]	00000000	01010011	01111111	11111011	10010011	01111001	11111111	11111111
[768, 831]	00000001	00000000	01110011	11111111	00110001	00010011	01110101	11111111
[832, 895]	00000000	01000001	10110111	11101110	00000001	00100001	11111100	11111111
[896, 959]	00000000	10010011	01110101	11111111	00010001	01101011	11110101	11111111
[960, 1023]	11101001	11110111	11111111	11111011	10110111	11111111	11111011	11111111
[1024, 1087]	00000001	00100011	00000001	00111111	00010001	00000001	01110111	11111111
[1088, 1151]	00000001	01110101	01101011	01111111	00000000	01010011	11111110	11111111
[1152, 1215]	00000001	01100001	01111111	11111111	00101001	00110111	11111111	11111111
[1216, 1279]	00000000	01110011	00111111	01111011	10010010	01111101	11111111	11111111
[1280, 1343]	00000001	00000000	01111011	11111110	00101111	00011011	01111111	11111111
[1344, 1407]	00000000	01000001	00110111	11111110	00001001	00110111	01111110	01111111
[1408, 1471]	00000000	11010010	01111111	11111111	00011011	01101111	11111111	11111111
[1472, 1535]	00000000	01110101	01111111	01110111	00100111	01111111	01111011	01111111
[1536, 1599]	00000001	00000011	00000001	00001001	00010001	00000001	01000001	10010011
[1600, 1663]	00000001	01110101	00100001	01010101	00000000	01010001	10000000	11110111
[1664, 1727]	00000001	01000001	01101011	00010011	00000001	00000000	11111011	11111111
[1728, 1791]	00000000	01010001	00000001	01110011	00000000	01000001	10110111	11111111
[1792, 1855]	00000001	00000000	01100001	10000001	00100111	00001001	00011110	10111111
[1856, 1919]	00000000	01000000	00000001	01010110	00001000	00000000	00010000	01111111
[1920, 1983]	00000000	10000000	00100001	01110111	00000011	00000001	00111111	11111111
[1984, 2047]	01101000	11010000	11110011	10110011	00000000	11010011	11111011	11111111
[2048, 2111]	00000001	00000011	00110111	11111111	00110011	00110111	01111111	11111111
[2112, 2175]	00000001	01110111	01111111	11111111	00010001	01111011	11111111	11111111
[2176, 2239]	00000001	11110111	01111111	11111111	00111111	11111111	11111101	11111111
[2240, 2303]	00010010	11110111	11111111	11111111	11111111	11111101	11111111	01111111
[2304, 2367]	00000001	00010010	01111101	11111111	00111111	01111111	11111111	11111111
[2368, 2431]	00000000	01100010	11111111	01111111	00111111	01111111	11111111	11111111
[2432, 2495]	00010000	11111111	11110111	11111111	01111111	11111111	01111111	11111111
[2496, 2559]	11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
[2560, 2623]	00000001	00100011	00000001	00011011	00010001	00100011	01110111	11111111
[2624, 2687]	00000001	01110101	00101011	01110111	00000000	01000001	10111110	11111111
[2688, 2751]	00000001	11000001	01011011	01111111	00001001	00110011	01111101	11111111
[2752, 2815]	00000000	01010001	00110111	11111011	10101001	10110001	11111111	11111111
[2816, 2879]	00000001	00000000	01110001	10110111	00100001	00000011	01110101	11111111
[2880, 2943]	00000000	01000000	00010111	01101111	00000000	00000001	01111101	11111111
[2944, 3007]	00000000	11000001	01110101	11111111	00000001	10101011	01010001	11111111
[3008, 3071]	11101000	11010011	11111111	11111011	10111011	11111111	11111011	11111111
[3072, 3135]	00000001	00100011	00000001	00011011	00110001	00000001	01010011	01111111
[3136, 3199]	00000001	01110101	00101001	01111111	00000000	01010001	10110110	11111111
[3200, 3263]	00000001	11100000	01111011	11111111	00001010	00111011	01111111	11111111
[3264, 3327]	00000000	01110001	01111111	11111011	10001000	01110101	11111111	01111111
[3328, 3391]	00000001	00000000	01100001	11110110	00111111	00001001	01111111	11111111
[3392, 3455]	00000000	01000000	00010111	01111111	00001000	00010011	01111110	01111111
[3456, 3519]	00000000	10000000	01110111	11111111	00101011	00101111	01111111	01111111
[3520, 3583]	00000000	01110001	01111111	01110111	00101011	01111111	00111011	01111111
[3584, 3647]	00000001	00000011	00000001	00001001	00010001	00000001	01000001	00000001
[3648, 3711]	00000001	01110101	00100001	01010101	00000000	01010001	10000000	01010011
[3712, 3775]	00000001	01000001	01001001	00000001	00001001	00000000	00000001	00010011
[3776, 3839]	00000000	01010001	00000000	01010011	10000000	01000001	00010011	01111111
[3840, 3903]	00000001	00000000	01100001	10000000	00100001	00000001	00000001	00010011
[3904, 3967]	00000000	01000000	00000000	01000000	00000000	00000000	00000000	00010011
[3968, 4031]	00000000	10000000	00000000	00010011	00000001	00000001	01010001	01111111
[4032, 4095]	00000000	01010000	00000000	01110011	00000001	01010100	00110001	01110111

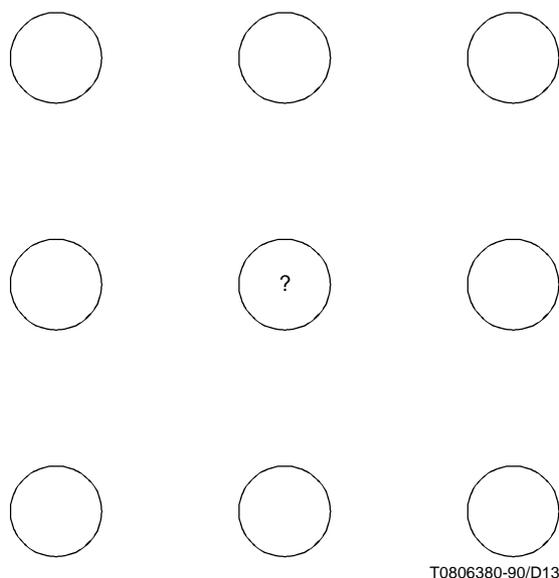


Figura 5 – Definición de la vecindad 8

La Figura 7 muestra un par de líneas de alta resolución y una línea de baja resolución asociada. También se muestra la ubicación virtual empleada para codificar el valor LNTTP.

Como sugiere esta figura, el valor LNTTP del seudopíxel será codificado por el codificador aritmético antes de que se codifiquen cualquiera de los píxels de alta resolución ordinarios. Al codificar este seudopíxel, TPVALUE y DPVALUE serán siempre 2. El contexto CX que se utilizará para codificar es el mismo contexto que se utiliza para codificar un píxel ordinario de fase 3, y está rodeado de píxels como se indica en la Figura 8. En esta figura, «F» significa primer plano y «B» significa fondo.

NOTA – Este contexto concreto se eligió para su reutilización como texto para codificar LNTTP, ya que se presenta infrecuentemente, y en la mayoría de las imágenes la probabilidad de codificar negro dentro del mismo es pequeña, al ser la probabilidad de que LNTTP sea igual a uno. La alternativa más evidente de codificar LNTTP en su propio contexto aumenta desgraciadamente el número de contextos justamente por encima de una potencia de 2.

La Figura 9 muestra el procesamiento necesario para producir la señal de salida TPVALUE. Dicho en palabras, si LNTTP es igual a cero y el píxel de baja resolución asociado con un píxel de alta resolución PIX es del mismo color que todos los píxels de su vecindad 8, entonces TPVALUE es igual a ese color. En otro caso, se pone igual a dos para indicar que no puede efectuarse una predicción.

6.4.2 Procesamiento en un decodificador

Al comienzo de cada par de líneas de alta resolución, se decodificará el seudopíxel LNTTP de la TP (véase la Figura 7). Al decodificar LNTTP, las entradas TPVALUE y DPVALUE serán iguales a 2 y CX será el que se indica en 6.4.1.

Al decodificar un determinado píxel de alta resolución PIX se generará TPVALUE como en un codificador.

6.5 Predicción típica de capa de resolución mínima

La TP de la capa de resolución mínima puede ser activada o neutralizada con el bit TPBON en el campo de opciones de BIH. Si se neutraliza (TPBON = 0), el bloque TPB de un codificador o un decodificador producirá simplemente TPVALUE igual a 2 para todos los píxels, indicando así a los bloques codificador y decodificador que no se está haciendo predicción. Además, cuando se neutraliza TP de capa de resolución mínima, el seudopíxel SLNTP no será codificado ni decodificado por el codificador aritmético. En el resto de este punto se supone que TP de capa de mínima resolución está activada (TPBON = 1).

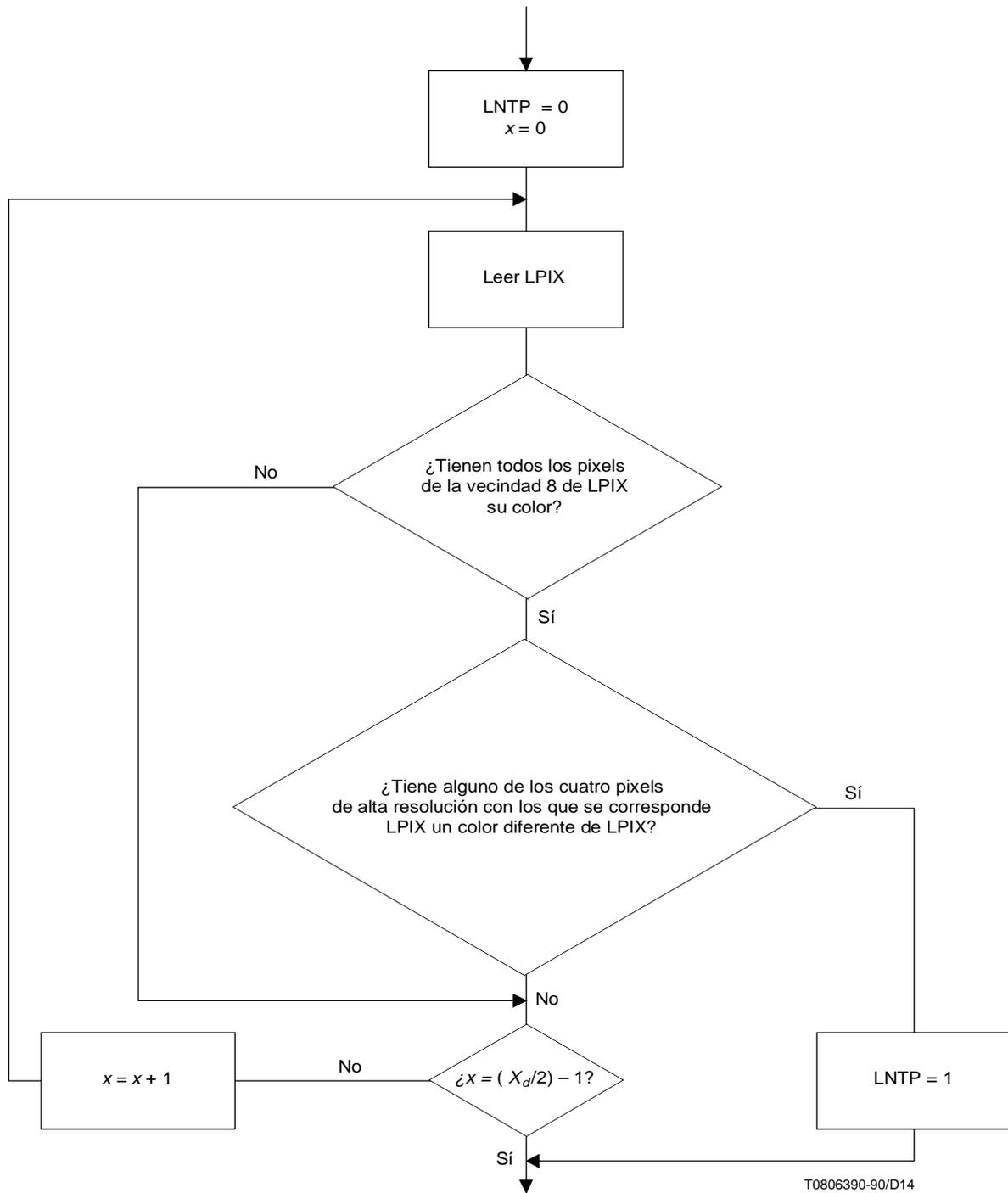


Figura 6 – Procesamiento para determinar LNTP

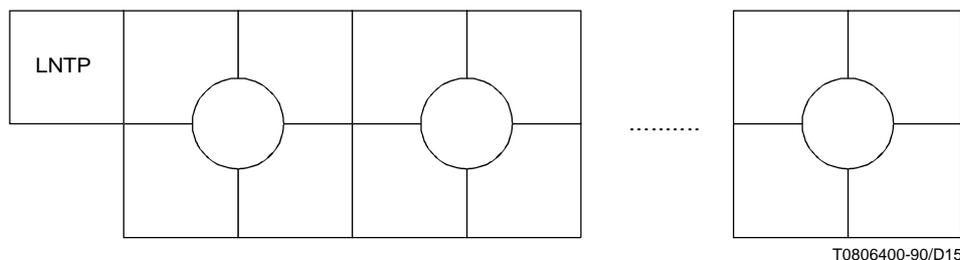


Figura 7 – Posición de un pseudopixel con relación a los pixels ordinarios

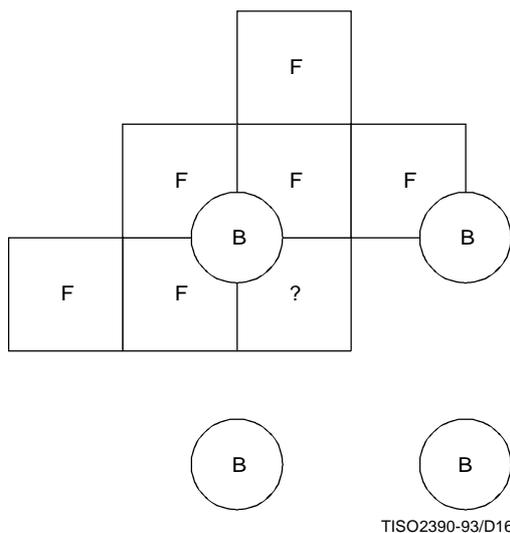


Figura 8 – Contexto reutilizado para codificar el pseudopixel en TP de capa diferencial

6.5.1 Procesamiento por el codificador

Sea y la línea vigente. Si difiere en cualquier posición de pixel de la línea anterior, $LNTP_y$ será entonces igual a 1, y la línea vigente se dice que es no típica. En otro caso, $LNTP_y$ será igual a 0. Al determinar si la primerísima línea de una imagen es no típica se supondrá, como de costumbre, que la línea inmediatamente por encima de la imagen es del color del fondo.

NOTA – Mientras que casi todas las líneas son «típicas» en el sentido de la TP de capa diferencial, sólo una modesta fracción son «típicas» en el sentido de la TP de capa de mínima resolución.

Definamos

$$SLNTP_y = !(LNTP_y \oplus LNTP_{y-1}) \tag{5}$$

donde el símbolo \oplus designa la operación exclusivo-o y el símbolo $!$ designa negación lógica. En otras palabras, $SLNTP_y$ será igual a 1 si y sólo si $LNTP_y$ es el mismo que era en la línea anterior. Para la línea superior de una imagen, $LNTP_{y-1}$ se pondrá igual a 1.

Cuando se activa la TP de capa de resolución mínima un pseudopíxel igual en valor a $SLNTP$ se codificará antes de que se codifiquen píxeles de la línea y (véase la Figura 10).

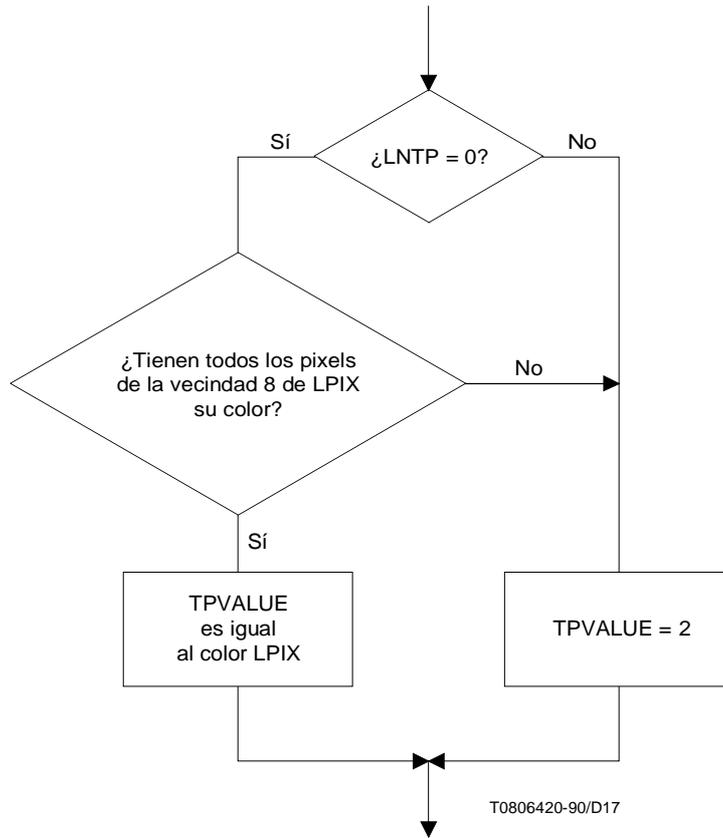
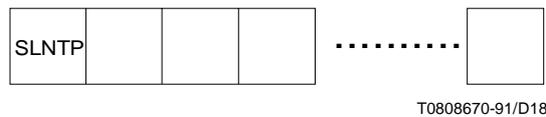


Figura 9 – Procesamiento para determinar TPVALUE



T0808670-91/D18

Figura 10 – Posición de un pseudopíxel con relación a los píxeles ordinarios

Cuando se codifica $SLNTP$, se codificará en el contexto presentado en la Figura 11 si $LRLTWO$ es 0, y en el presentado en la Figura 12 si $LRLTWO$ es 1 (véase 6.7.1). En esta figura, «F» significa primer plano y «B» significa fondo. Al codificar $SLNTP$, $TPVALUE$ será siempre igual a 2. Es decir, nunca puede predecirse $SLNTP$ con la TP, y debe codificarse siempre aritméticamente.

NOTA – Codificar aritméticamente cambios en $LNTP$ es más eficaz que codificar aritméticamente $LNTP$. En la TP de capa de resolución mínima $LNTP$ no toma ni el valor 1 ni el valor 0 con gran probabilidad, y no puede codificarse en entropía con gran eficacia.

Si $LNTP_y$ es igual a 0, el bloque TPB producirá como $TPVALUE$ el valor común al píxel actual y al píxel que está inmediatamente por encima. En otro caso, producirá 2 para indicar que no puede hacerse predicción.

6.5.2 Procesamiento por el decodificador

Si TPBON es igual a uno, se decodificará el indicador de mismidad SLNTP (véase la Figura 10). Al decodificar SLNTP, TPVALUE será 2, y CX será el de la Figura 11 ó Figura 12, según convenga.

		B	B	F	
F	F	B	B	F	
B	F	?			TISO2400-93/D19

Figura 11 – Contexto reutilizado para codificar el seudopixel en TP de capa de resolución mínima (plantilla de tres líneas)

		B	F	F	B	B	F
B	F	B	F	?			

TISO2410-93/D20

Figura 12 – Contexto reutilizado para codificar el seudopixel en TP de capa de resolución mínima (plantilla de dos líneas)

El decodificador creará de nuevo LNTP_y mediante

$$LNTP_y = !(SLNTP_y \oplus LNTP_{y-1}) \tag{6}$$

Como ocurre en el codificador, esta iteración será inicializada con LNTP puesta igual a 1 para la línea inmediatamente por encima de la línea superior de la imagen real.

Si LNTP_y es igual a 0, el bloque TPB producirá el valor del pixel inmediatamente por encima del vigente. En otro caso, producirá 2 para indicar que no puede hacerse predicción.

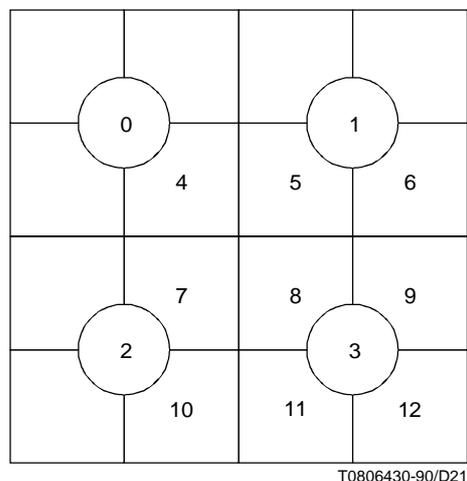
6.6 Predicción determinística (DP)

La DP será activada o neutralizada con el bit DPON en el campo opciones de BIH. Si se neutraliza DP (DPON = 0), el bloque DP de un codificador o de un decodificador producirá simplemente DPVALUE = 2 para todos los pixels. En el resto de esta subcláusula y sus subcláusulas se supone que DP está activado (DPON = 1).

Si se utiliza DP al codificar una imagen, se supondrá que se utilizaron las tablas DP descritas en esta subcláusula para hacer predicciones, a menos que se haya señalado la utilización de tablas DP privadas, como se indica en 6.2.

6.6.1 Definición de pixels asociados

Para los fines de descripción del algoritmo de predicción determinística, la Figura 13 muestra la designación que se utilizará para los pixels necesarios procedentes de las imágenes de baja resolución y alta resolución. Siempre que se haga referencia a un pixel que, debido a los efectos de bordes, no esté realmente en la barra vigente, el valor de este pixel será determinado por las reglas de bordes generales de 6.1.2.



T0806430-90/D21

Figura 13 – Designación de los pixels utilizados por DP

6.6.2 Tablas DP por defecto

Los pixels vecinos, o «de referencia», que se utilizan para hacer predicciones para cada fase especial concreta serán los que se indican en el Cuadro 18. Obsérvese que para cada una de las cuatro fases espaciales posibles se utiliza un conjunto diferente de pixels para efectuar predicciones DP. Los pixels utilizados en cada posible fase son los que se designan en la Figura 13, y son conocidos tanto por un codificador como por un decodificador en el momento en que debe codificarse la fase espacial concreta. Además, en este cuadro figura un número que indica cuántas combinaciones de pixels de referencia dan lugar realmente a una predicción (o acierto) DP, cuando se utilizan las reglas DP que siguen.

Cuadro 18 – Pixels DP para cada fase espacial

Fase	Pixel objetivo	Pixels de referencia	Número de saltos con reducción de resolución por defecto
0	8	0, 1, 2, 3, 4, 5, 6, 7,	20
1	9	0, 1, 2, 3, 4, 5, 6, 7, 8	108
2	11	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10	526
3	12	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	1044

Las tablas DP privadas no utilizarán para ninguna de las cuatro fases pixels distintos de los indicados en el Cuadro 18. El número de patrones de pixels de referencia que «acertarán» será en general distinto de los números indicados en el Cuadro 18 para el algoritmo de reducción de resolución por defecto.

Los Cuadros 19, 20, 21 y 22 definen DP para el algoritmo de reducción de resolución por defecto. Estos cuatro cuadros deben utilizarse respectivamente para determinar DPVALUE en cada una de las cuatro fases especiales 0, 1, 2 y 3. El índice que se lleva al cuadro se crea del mismo modo que el índice que se lleva al Cuadro 17 de reducción de resolución, con la diferencia de que la significación de los bits será definida por los números de pixel indicados en la Figura 13 y no en la Figura 4.

Las entradas a estos cuadros dan DPVALUE y son todas 0, 1 ó 2. Un «2» indica que no es posible hacer una predicción determinística. Un «0» indica que hay un acierto DP, y que el pixel objetivo debe ser del color de fondo (0). Un «1» indica que hay un acierto DP, y que el pixel objetivo debe ser del color del primer plano (1). Como en el Cuadro 17, las entradas se leen de izquierda a derecha con índice creciente.

Cuadro 19 – Tabla DP para predecir la fase espacial 0

Índice	DPVALUE							
[0, 63]	02222222	22222222	22222222	22222222	02222222	22222222	22222222	22222222
[64, 127]	02222222	22222222	22222222	22222222	00222222	22222222	22222222	22222222
[128, 191]	02222222	22222222	00222222	22222222	02020222	22222222	02022222	22222222
[192, 255]	00222222	22222222	22222222	22222221	02020022	22222222	22222222	22222222

Cuadro 20 – Tabla DP para predecir la fase espacial 1

Índice	DPVALUE							
[0, 63]	22222222	22222222	22222222	22000000	02222222	22222222	00222222	22111111
[64, 127]	22222222	22222222	22222222	21111111	02222222	22111111	22222222	22112221
[128, 191]	02222222	22222222	02222222	22222222	00222222	22222200	20222222	22222222
[192, 255]	02222222	22111111	22222222	22222102	11222222	22222212	22220022	22222212
[256, 319]	20222222	22222222	00220222	22222222	20000222	22222222	00000022	22222221
[320, 383]	20222222	22222222	11222222	22222221	22222222	22222221	22221122	22222221
[384, 447]	20020022	22222222	22000022	22222222	20202002	22222222	20220002	22222222
[448, 511]	22000022	22222222	00220022	22222221	21212202	22222222	22220002	22222222

Cuadro 21 – Tabla DP para predecir la fase espacial 2

Índice	DPVALUE							
[0, 63]	22222222	12222222	22222222	22222222	02222222	12222222	02222222	11222222
[64, 127]	22222222	22222222	02222222	12222222	02222222	11222222	22221122	22222222
[128, 191]	00202222	11111111	00200222	11111111	00222222	21122222	10222222	22111222
[192, 255]	02222222	11222222	00222222	21222222	22222222	22202220	22220022	22112222
[256, 319]	20222222	21222222	20020022	22222222	20000222	22222220	22000022	22222212
[320, 383]	20220222	22211111	22020222	22112122	22000022	22122122	22002222	22222222
[384, 447]	20020022	22222200	22000022	22222212	22202022	22222222	20202202	22222212
[448, 511]	22202022	22222200	00002022	22222212	22222202	22222221	22002220	22212221
[512, 575]	02222222	22111122	02222222	11222222	22212122	22220000	22122122	22202000
[576, 639]	00000000	22222222	02222222	22000000	22002222	22222222	22002112	22222222
[640, 703]	20222222	21221122	20222222	22121222	22000022	22112122	02222222	22212222
[704, 767]	20222222	22222022	00022222	21111222	02000022	22222200	22002212	22222222
[768, 831]	22020222	22111122	22222022	22222200	22222022	22222212	22222202	22222221
[832, 895]	22000022	22222222	00201222	22222220	22022202	22222222	22002200	22222222
[896, 959]	22202222	22222222	22222202	22222221	22222222	22222221	22202220	22222222
[960, 1023]	22222222	22222222	22002202	22222221	20202220	22222222	22002220	22222222
[1024, 1087]	22222222	11111111	22222222	11111111	02222222	21111111	22222222	22222222
[1088, 1151]	22222222	11111111	02222222	22222222	02222222	21222222	22222222	22222002
[1152, 1215]	00222222	11111111	22222202	22221121	12222222	22222212	22002202	22221111
[1216, 1279]	02222222	21222222	22222222	22201202	22220222	22101222	22000022	22222221
[1280, 1343]	20222222	22112111	11020222	22211111	22202002	22222222	00202222	22222212
[1344, 1407]	22020022	22222211	22000022	22222212	22222022	22222212	22222202	22222212
[1408, 1471]	22002022	22211211	22222212	22222221	21222102	22222221	22222222	22222121
[1472, 1535]	22121122	22222222	22111122	22220222	22222200	22222221	22000002	22222221
[1536, 1599]	00000000	11111111	02222222	21222222	20222222	22121111	22220222	22121122
[1600, 1663]	00222222	22222222	22222222	21121222	20020222	22111111	22220022	22212122
[1664, 1727]	20220022	22111111	22020022	22221121	22000022	22222122	22220022	22222211
[1728, 1791]	20020222	22111111	22002222	22211122	11122222	22222111	22222202	22222210
[1792, 1855]	22200022	22222222	22122022	22222212	22222202	22222211	22222200	22222221
[1856, 1919]	22222022	22222222	22121222	22222222	22000000	22222211	22000000	22222211
[1920, 1983]	22222202	22222211	22222220	22222221	22222220	22222221	22222222	22222222
[1984, 2047]	22222200	22222221	22222220	22222221	22222222	22222222	22222220	22222222

Cuadro 22 – Tabla DP para predecir la fase espacial 3

Índice	DPVALUE								
[0, 63]	22222222	22222222	22222222	22222222	22222222	22222222	22222222	22222222	22222222
[64, 127]	22222222	22222222	22222222	22222222	22222222	22222222	22222202	22222212	22222212
[128, 191]	22222222	22222222	22222222	22222222	02222222	12222222	20222222	21222222	21222222
[192, 255]	22222222	22222222	02222222	12222222	22111121	22000020	22221122	22220022	22220022
[256, 319]	22222222	22222222	20000000	21111111	20000000	21111111	22000022	22111122	22111122
[320, 383]	20022222	21122222	22221222	22220222	22200222	22211222	22002222	22112222	22112222
[384, 447]	20000000	21111111	22000022	22111122	22202022	22212122	20202020	21212121	21212121
[448, 511]	22212111	22202000	00002022	11112122	22222212	22222202	22022222	22122222	22122222
[512, 575]	02222222	12222222	22222222	22222222	22020200	22121211	22211211	22200200	22200200
[576, 639]	00000000	11111111	00000000	11111111	22000000	22111111	22002220	22112221	22112221
[640, 703]	22222222	22222222	20222222	21222222	22221222	22220222	02020122	12121022	12121022
[704, 767]	20000000	21111111	02222222	12222222	02000100	12111011	22002220	22112221	22112221
[768, 831]	22222222	22222222	22222111	22222000	22222022	22222122	22222202	22222212	22222212
[832, 895]	22000000	22111111	00202000	11212111	22022200	22122211	22002210	22112201	22112201
[896, 959]	22202212	22212202	22222212	22222202	22222222	22222222	22202220	22212221	22212221
[960, 1023]	22222220	22222221	22002202	22112212	20202220	21212221	22002220	22112221	22112221
[1024, 1087]	22222222	22222222	22222222	22222222	02222222	12222222	02222222	12222222	12222222
[1088, 1151]	22222222	22222222	02222222	12222222	02222222	12222222	22221112	22220002	22220002
[1152, 1215]	22222222	22222222	22222202	22222212	22111121	22000020	22112222	22002222	22002222
[1216, 1279]	02222222	12222222	22112212	22002202	22212122	22202022	22010122	22101022	22101022
[1280, 1343]	20220222	21221222	22122222	22022222	22212111	22202000	00200022	11211122	11211122
[1344, 1407]	22111122	22000022	22000022	22111122	22222022	22222122	22222222	22222222	22222222
[1408, 1471]	22022022	22122122	22220022	22221122	22222212	22222202	22202222	22221222	22221222
[1472, 1535]	22202222	22212222	22222222	22222222	22222202	22222212	22111112	22000002	22000002
[1536, 1599]	22222222	22222222	02222222	12222222	20222222	21222222	22222222	22222222	22222222
[1600, 1663]	00000000	11111111	22222222	22222222	20222222	21222222	22020222	22121222	22121222
[1664, 1727]	20222222	21222222	22112212	22002202	22010222	22101222	22221122	22220022	22220022
[1728, 1791]	21222222	20222222	22022222	22122222	22201222	22210222	22222220	22222221	22222221
[1792, 1855]	22211111	22200000	22202022	22212122	22222222	22222222	22222202	22222212	22222212
[1856, 1919]	22222000	22222111	22222202	22222212	22111122	22000022	22001122	22110022	22110022
[1920, 1983]	22222222	22222222	22222222	22222222	22222222	22222222	22222222	22222222	22222222
[1984, 2047]	22222202	22222212	22222222	22222222	22222222	22222222	22222220	22222221	22222221
[2048, 2111]	02222222	12222222	22222222	22222222	22222222	22222222	20222222	21222222	21222222
[2112, 2175]	22222222	22222222	22222222	22222222	20222222	21222222	22221112	22220002	22220002
[2176, 2239]	22020000	22121111	22122111	22022000	22122222	22022222	02010222	12101222	12101222
[2240, 2303]	20222222	21222222	22222222	22222222	22020122	22121022	22112222	22002222	22002222
[2304, 2367]	22222222	22222222	22212211	22202200	22222012	22222102	22222212	22222202	22222202
[2368, 2431]	22112111	22002000	22202022	22212122	22222222	22222222	22222222	22222222	22222222
[2432, 2495]	22202222	22212222	22222202	22222212	22222222	22222222	22222020	22222121	22222121
[2496, 2559]	22222222	22222222	22222202	22222212	22222220	22222221	22222222	22222222	22222222
[2560, 2623]	22111122	22000022	20222222	21222222	22112222	22002222	22020222	22121222	22121222
[2624, 2687]	22222222	22222222	21222222	20222222	22220000	22221111	22222000	22222111	22222111
[2688, 2751]	22221122	22220022	22020222	22121222	22222222	22222222	22111122	22000022	22000022
[2752, 2815]	22000200	22111211	22201222	22210222	22222222	22222222	22222200	22222211	22222211
[2816, 2879]	22202022	22212122	22222222	22222222	22222202	22222212	22222220	22222221	22222221
[2880, 2943]	22222200	22222211	22221102	22220012	22222220	22222221	22222222	22222222	22222222
[2944, 3007]	22222202	22222212	22222220	22222221	22222220	22222221	22222222	22222222	22222222
[3008, 3071]	22222220	22222221	22222220	22222221	22222222	22222222	22222222	22222222	22222222
[3072, 3135]	00000000	11111111	00000000	11111111	20000000	21111111	02222222	12222222	12222222
[3136, 3199]	00000000	11111111	22222222	22222222	21222222	20222222	22220222	22221222	22221222
[3200, 3263]	22000000	22111111	22220020	22221121	02010000	12101111	22220020	22221121	22221121
[3264, 3327]	20222222	21222222	22020222	22121222	22122022	22022122	22222220	22222221	22222221
[3328, 3391]	22000000	22111111	00202000	11212111	22222220	22222221	22220002	22221112	22221112
[3392, 3455]	22202200	22212211	22222202	22222212	22222202	22222212	22222222	22222222	22222222
[3456, 3519]	22220200	22221211	22220010	22221101	20222020	21222121	22220020	22221121	22221121
[3520, 3583]	22111122	22000022	22112022	22002122	22222222	22222222	22222220	22222221	22222221
[3584, 3647]	22222222	22222222	21222222	20222222	22020000	22121111	22122022	22022122	22022122
[3648, 3711]	22000000	22111111	21020222	20121222	22202000	22212111	22002222	22112222	22112222
[3712, 3775]	22002200	22112211	22202200	22212211	22222222	22222222	22222200	22222211	22222211
[3776, 3839]	22202000	22212111	22220022	22221122	00000000	11111111	22222222	22222222	22222222
[3840, 3903]	22222200	22222211	22112202	22002212	22222220	22222221	22222222	22222222	22222222
[3904, 3967]	22222200	22222211	22121212	22020202	22222222	22222222	22222222	22222222	22222222
[3968, 4031]	22222220	22222221	22222222	22222222	22222222	22222222	22222222	22222222	22222222
[4032, 4095]	22222222	22222222	22222222	22222222	22222222	22222222	22222222	22222222	22222222

6.6.3 Formato de la tabla DP

Si $DPON = 1$, $DPPRIV = 1$ y $DPLAST = 0$, la tabla DP privada se codificará al campo DPTABLE de BIH (véase 6.2). El campo DPTABLE se definirá comparando la estructura de las cuatro tablas anteriores que definen DP para el algoritmo de reducción de resolución por defecto. En particular, será una concatenación de las cuatro tablas con dos bits atribuidos a cada entrada a la tabla de forma que las cuatro entradas compongan un byte. Las entradas tipográficamente situadas más a la izquierda y más arriba en los cuadros compondrán los bits de orden superior de los bytes y los primeros bytes del campo DPTABLE. Cuando no es posible hacer una predicción DP, se codificará 2 en el campo de dos bits.

NOTA – Este permiso se concede de manera que el bit más significativo del campo de dos bits resulte por sí mismo una indicación de que no puede hacerse una predicción DP.

Como existen 8, 9, 11 y 12 pixels de referencia para predecir respectivamente los pixels de las fases espaciales 0, 1, 2 y 3, el campo DPTABLE tendrá

$$1728 = 2 \times (256 + 512 + 2048 + 4096) / 8 \tag{7}$$

bytes de longitud.

6.7 Plantillas modelo y plantillas adaptativas

Las plantillas modelo definen una vecindad en torno a un pixel que ha de codificarse. Los valores de los pixels de estas vecindades, más la fase espacial de las capas diferenciales, definen un contexto, con un adaptador de codificación aritmética separado utilizado para cada contexto diferente (véase 6.8). Aunque una plantilla es un patrón geométrico de pixels, los pixels de una plantilla se dice que asumen valores cuando la plantilla está alineada a una parte determinada de la imagen.

6.7.1 Capa de resolución mínima

La Figura 14 muestra la plantilla que se utilizará cuando se codifique la capa de resolución mínima y cuando LRLTWO es 0.

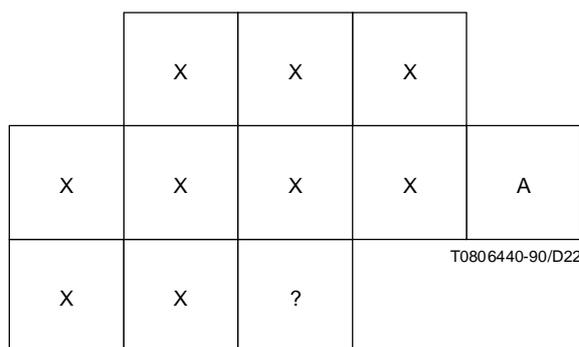


Figura 14 – Plantilla modelo para la capa de resolución mínima

El pixel designado por «?» corresponde a un pixel que ha de codificarse y no forma parte de la plantilla. Los pixels designados por «X» corresponden a pixels ordinarios de la plantilla, y el pixel designado por «A» es un pixel especial de la plantilla denominado pixel «adaptativo» o pixel «AT». Este pixel tiene la particularidad de que se admite que su posición cambie durante el proceso de codificación de una imagen. Véase en 6.7.3 una descripción de los pixels AT. La «A» indica la posición inicial del pixel AT.

Los valores de los pixels de la plantilla se combinarán para formar un contexto. Cada pixel de la plantilla modelo (incluido el pixel adaptativo) se corresponderá con un bit específico del contexto, aunque los pixels de la plantilla pueden asignarse a bits del contexto en cualquier orden. Dado que existen 10 pixels en esta plantilla, los contextos asociados con la capa de resolución mínima pueden asumir 1024 valores diferentes. Este contexto se utilizará para identificar qué adaptador codificador aritmético debe utilizarse para codificar el pixel a codificar, como se indica en el 6.8

Si $LRLTWO$ es 1, la plantilla modelo de la capa de resolución mínima será la presentada en la Figura 15.

	X	X	X	X	X	A
X	X	X	X	?	T0808700-91/D23	

Figura 15 – Plantilla modelo de dos líneas para la capa de resolución mínima

El significado de «X», «A» y «?» es el mismo que antes.

NOTA – La velocidad de ejecución de soporte lógico de la capa de resolución mínima será generalmente algo más rápida con la plantilla de dos líneas que con la plantilla de tres líneas. La penalización por utilizar la plantilla de dos líneas es de una pérdida aproximada del 5% en la eficacia de compresión.

Siempre que cualquiera de los pixels de la plantilla de la Figura 14 ó 15 caiga fuera de los límites de la imagen o barra, se aplicarán las reglas de bordes generales de 6.1.2.

6.7.2 Capa diferencial

La Figura 16 muestra las plantillas que se utilizarán al codificar imágenes de capa diferencial. Obsérvese que esta plantilla contiene referencia a pixels de la imagen de resolución inmediatamente inferior así como a pixels de la imagen que se codifica, y que la plantilla modelo es diferente para fases diferentes. Los símbolos «?», «X» y «A» tienen el mismo significado que en la subcláusula anterior.

Los contextos se formarán a partir de estas plantillas de modo similar a lo que se describió para la plantilla de capa de resolución mínima. Cada pixel de una plantilla contribuirá con un bit al contexto. Además, al codificar imágenes de capa diferencial, se añadirán dos bits adicionales al contexto para describir la fase del pixel que se codifica. Como antes, pueden utilizarse cualesquiera bits para describir la información de fase, aunque la asignación de pixels y de información de fase a bits del contexto permanecerá fija mientras se codifica una imagen. Dado que existen diez pixels en las plantillas de capa diferencial y que se utilizan dos para describir la información de fase, existen 4096 posibles contextos diferentes mientras se procesan imágenes diferenciales. Este contexto se utilizará para identificar qué adaptador codificador aritmético debe utilizarse para codificar el pixel considerado.

6.7.3 Pixels de plantilla adaptativa

Al codificar las capas diferenciales, así como la capa de resolución mínima, se permitirá que la plantilla modelo cambie en el modo restringido descrito en este punto.

El pixel de la plantilla modelo autorizado a cambiar se denominará pixel AT. Su posición inicial (o por defecto) es indicada por «A» en las Figuras 14, 15 y 16, respectivamente, para la codificación de capa de resolución mínima con una plantilla de tres líneas, codificación de capa de resolución mínima con una plantilla de dos líneas, y codificación de capa diferencial. En general, el pixel AT puede desplazarse independientemente para todas las capas a cualquier punto del campo mostrado en la Figura 17.

Sin embargo, la nueva posición AT no se superpondrá con pixels ordinarios de la plantilla. (De aquí que los desplazamientos admisibles para la codificación de capa de resolución mínima de dos líneas, codificación de capa de resolución mínima de tres líneas, y codificación de capa diferencial sean ligeramente diferentes.)

Un desplazamiento AT de capa diferencial es eficaz para las cuatro fases simultáneamente. Si existe más de una capa diferencial, los desplazamientos en cada una son independientes; ahora bien, en cualquiera de las capas una vez que el pixel AT se ha desplazado para una barra determinada, las barras sucesivas seguirán utilizando la nueva posición.

Los parámetros M_X y M_Y definen el tamaño del rectángulo en la Figura 17. Los límites absolutos y el soporte mínimo sugerido son los indicados en los Cuadros 9 y A.1.

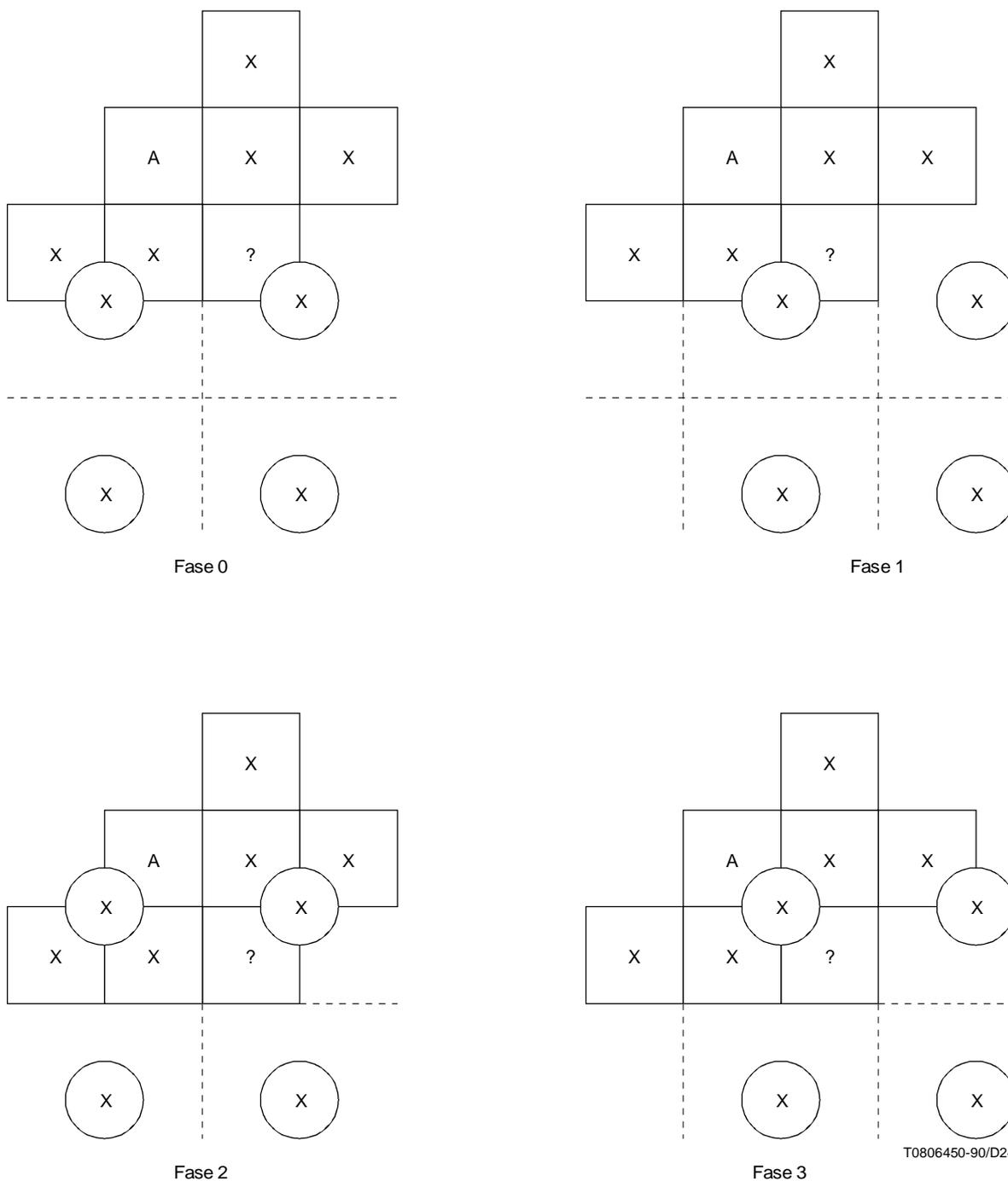


Figura 16 – Plantillas modelo para codificación de capa diferencial

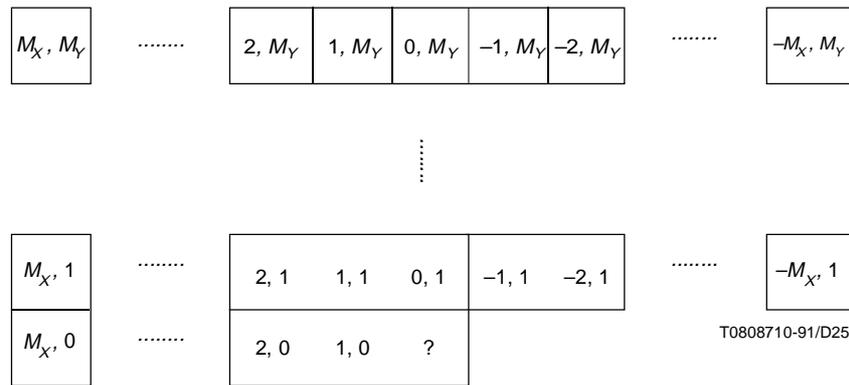


Figura 17 – Campo al que se restringen los desplazamientos AT

NOTA – Dado que generalmente es más costoso para el soporte físico soportar el desplazamiento vertical del pixel AT, el soporte mínimo sugerido para M_y se limita a cero de manera que se restrinja el desplazamiento del pixel AT a cualquier punto que no sea la posición por defecto a la línea vigente que se codifica.

Si un codificador desea cambiar la posición del pixel AT, informará del cambio al decodificador codificando τ_x , τ_y e y_{AT} como se indica en el Cuadro 14. Los números codificados a τ_x y τ_y serán, respectivamente, los desplazamientos horizontales y verticales a partir del pixel objetivo, como muestra la Figura 17. El número posiblemente negativo τ_x se codificará en forma de complemento a dos. El número codificado a y_{AT} será el número de la línea de alta resolución al comienzo de la cual se efectuará el cambio. La numeración de línea utilizada recomenzará por 0 en la parte superior de cada barra.

Es admisible desplazar el pixel AT hacia atrás a su posición inicial (o por defecto) después de haberlo desplazado una vez fuera de ella. La posición por defecto del pixel AT se codificará siempre $\tau_x = 0$ y $\tau_y = 0$ y no las coordenadas X e Y verdaderas.

NOTA – Este convenio es conveniente debido a que las coordenadas verdaderas son diferentes en la codificación de capa de resolución mínima y en la codificación de capa diferencial. También hace posible que un codificador informe a un decodificador que nunca habrá desplazamiento AT si se hacen M_x y M_y iguales a 0.

El Anexo C expone una técnica de cálculo simple para hacer determinaciones del momento en que es deseable un cambio del pixel AT y del lugar a que debe desplazarse.

6.8 Codificación aritmética

El codificador de entropía utilizado en esta Especificación es un codificador de compresión aritmética adaptativa. En esta subcláusula (incluidos todos los subpuntos) los diagramas de flujo y el Cuadro 24 son únicamente normativos, en el sentido de que definen una salida que las realizaciones alternativas deben duplicar. Toda la información básica y lo expuesto en los subpuntos de esta subcláusula es de índole informativa.

NOTA – Se pretende que las operaciones de codificación aritmética descritas en este punto sean idénticas a las operaciones de codificación aritmética descritas en la Rec. T.81 del CCITT | ISO/CEI 10918-1. No obstante, en caso de diferencia casual entre las descripciones, se utilizará el procedimiento descrito en esta Especificación.

Se permite tener hasta cuatro segmentos marcadores de $ATMOVE$ en cualquiera de las barras. Cuando haya varios segmentos marcadores de $ATMOVE$ en una barra, sus Y_{AT} de líneas efectivas distintos y ordenados de modo que los marcadores de $ATMOVE$ anteriores tengan valores de Y_{AT} numéricamente inferiores.

Para cada barra de cada capa de resolución, el codificador aritmético producirá un tren de bytes SCD. Tendrá como entradas cuatro trenes, cada uno de los cuales proporcionará un valor de cada pixel en la barra que se codifica. Como se muestra en la Figura 18, estas cuatro entradas serán un pixel PIX un valor de contexto CX , una indicación $TPVALUE$ de TP , y una indicación $DPVALUE$ de DP .



Figura 18 – Entradas y salidas del codificador

El pixel a codificar es generalmente un simple pixel de la imagen, pero si se activa TP de la capa de resolución mínima de la TP de capa diferencial, será ocasionalmente un seudopixel, LNTP (capa diferencial) o SLNTP (capa de resolución mínima). Las entradas CX, TPVALUE, y DPVALUE se generan como se indica en 6.7, 6.4, 6.5 y 6.6.

Para cada barra de cada capa de resolución, el decodificador aritmético leerá un tren de bytes SCD. Como se muestra en la Figura 19, este tren, así como los trenes de entradas por pixels CX, TPVALUE y DPVALUE se procesarán para crear de nuevo el tren PIX.

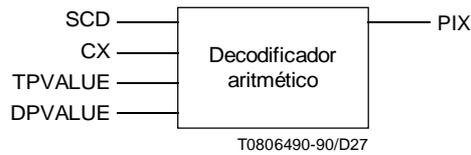


Figura 19 – Entradas y salidas del decodificador

Las entradas CX, TPVALUE y DPVALUE son idénticas a las utilizadas en el codificador.

Lo más simple es especificar los requisitos del codificador y del decodificador describiendo procedimientos de muestra. Existen muchos procedimientos equivalentes. Algunos aventajarán a otros en velocidad, utilización de memoria o sencillez. Algunos son más adecuados para su realización en soporte físico y otros más adecuados para su realización en soporte lógico. La elección tiene que sopesarse atendiendo a la máxima simplicidad y concisión. Pueden utilizarse cualesquiera procedimientos de codificación o decodificación que produzcan las mismas salidas que los procedimientos de muestra. Esta equivalencia de salida será el único requisito.

6.8.1 Conceptos fundamentales de codificación aritmética (informativo)

6.8.1.1 Subdivisión de intervalos

La subdivisión de intervalos de probabilidad recurrente de la codificación de Elias es la base de la codificación aritmética. Conceptualmente, una secuencia de entrada de símbolos se hace corresponder en un número real en el intervalo [0, 1], donde el corchete en un extremo de intervalo indica que se permite la igualdad, y un paréntesis que no se admite. Lo que se transmite o almacena en lugar de la secuencia original de símbolos es la expansión binaria de x .

La Figura 20 muestra un ejemplo de esta división de intervalos sirviéndose de una secuencia inicial 0,1, 0, 0 a codificar.

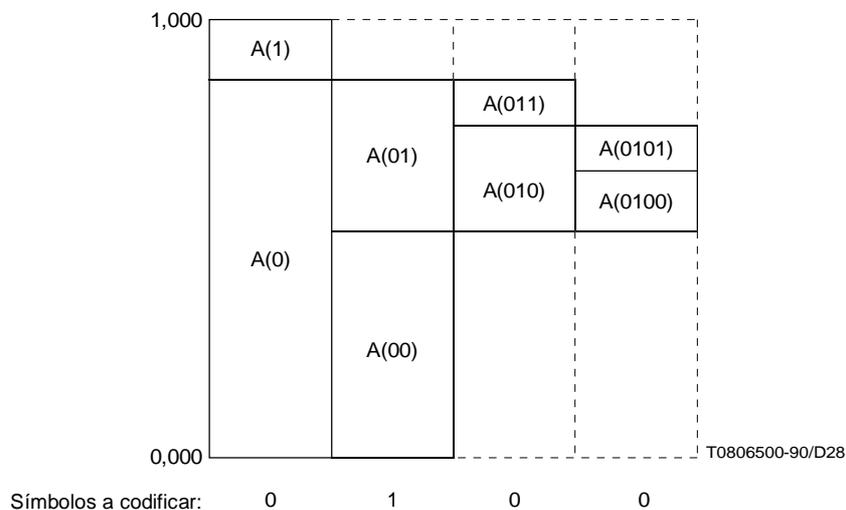


Figura 20 – Subdivisión de intervalos

La parte de $[0, 1]$ en la que se sabe que se halla x después de codificar una secuencia inicial de símbolos se conoce como el intervalo de codificación vigente. Para cada entrada binaria, el intervalo de codificación vigente se divide en dos subintervalos de tamaños proporcionales a las probabilidades relativas de las apariciones de valores de símbolo. El nuevo intervalo de codificación vigente se convierte en el asociado con el valor de símbolo efectivamente producido. En un codificador, el conocimiento del intervalo de codificación vigente se mantiene en una variable que da su tamaño y una segunda variable que da su base (límite inferior). El tren de salida se obtiene de la variable que apunta a la base.

En la partición del intervalo vigente en dos subintervalos, el subintervalo del símbolo menos probable (LPS) se ordena por encima del subintervalo del símbolo más probable (MPS). Por tanto, cuando se codifica el LPS, se añade a la base el subintervalo MPS. Este convenio de codificación exige que los símbolos se reconozcan como MPS o LPS, más que como 0 ó 1. Por consiguiente, el tamaño del intervalo LPS y el sentido del MPS para cada símbolo deben conocerse a fin de codificar ese símbolo.

Dado que el tren de código siempre apunta a un número real en el intervalo de codificación vigente, el proceso de decodificación consiste en determinar, para cada decisión, a qué intervalo apunta la cadena de código. Esto se hace también recurrentemente, utilizando el mismo proceso de subdivisión de intervalo que en el codificador. Cada vez que se decodifica una decisión, el decodificador sustrae todo intervalo que el codificador añadió al tren de código. Por tanto, el tren de código en el decodificador es un puntero al intervalo vigente relativo a la base del intervalo vigente.

Dado que el proceso de codificación exige la adición de fracciones binarias más que la concatenación de palabras de código enteras, las decisiones binarias más probables pueden a menudo codificarse a un costo mucho menor que un bit por decisión.

6.8.1.2 Convenios de codificación y aproximaciones (informativo)

Es posible realizar estas operaciones de codificación utilizando aritmética de enteros de precisión fija. Un registro A contiene el tamaño del intervalo de codificación vigente normalizado para que caiga siempre en la gama $[0x8000, 0x10000]$, donde un prefijo «0x» designa un entero hexadecimal. Siempre que de resultados de la codificación un símbolo A caiga temporalmente por debajo de $0x8000$, se dobla recurrentemente hasta que sea superior a $0x8000$. Estas duplicaciones se denominan «renormalizaciones».

Un segundo registro, C , contiene los bits traseros del tren de código. El registro C se dobla también cada vez que se dobla el tiempo A . Periódicamente (para evitar el desbordamiento de C), se suprime un byte de datos de los bits de alto orden del registro C , y se sitúa en una memoria intermedia de cadena de código externa. Las posibles transferencias deben resolverse antes de que el contenido de esta memoria intermedia se haya comprometido a salir.

Se utiliza una sencilla aproximación aritmética en la subdivisión de intervalos. Para un intervalo A y una estimación vigente p de la probabilidad LPS, un cálculo preciso del subintervalo LPS exigiría una multiplicación $p \times A$. En su lugar, se utiliza la aproximación

$$p \times A \approx p \times \bar{A} = \text{LSZ} \quad (8)$$

donde la tilde indica promedio a lo largo de la densidad de probabilidad de A y LSZ es una magnitud almacenada igual al tamaño del intervalo aproximado del LPS. Dado que A se mantiene en la gama $[0x8000, 0x10000]$, sustituyendo A por su media estadística no se introduce un error demasiado grande. Se comprueba empíricamente que A tiene una densidad de probabilidad inversamente proporcional a A .

Siempre que se codifique el LPS, el valor del subintervalo de MPS $A - \text{LSZ}$ se añade al registro de código y se reduce el intervalo de codificación al valor LSZ del subintervalo de LPS. Siempre que se codifique el MPS, el registro de código se deja invariable, y el intervalo se reduce a $A - \text{LSZ}$. Su A cae por debajo de $0x8000$ al realizar estas operaciones, se restablece a la gama adecuada renormalizando tanto A como C .

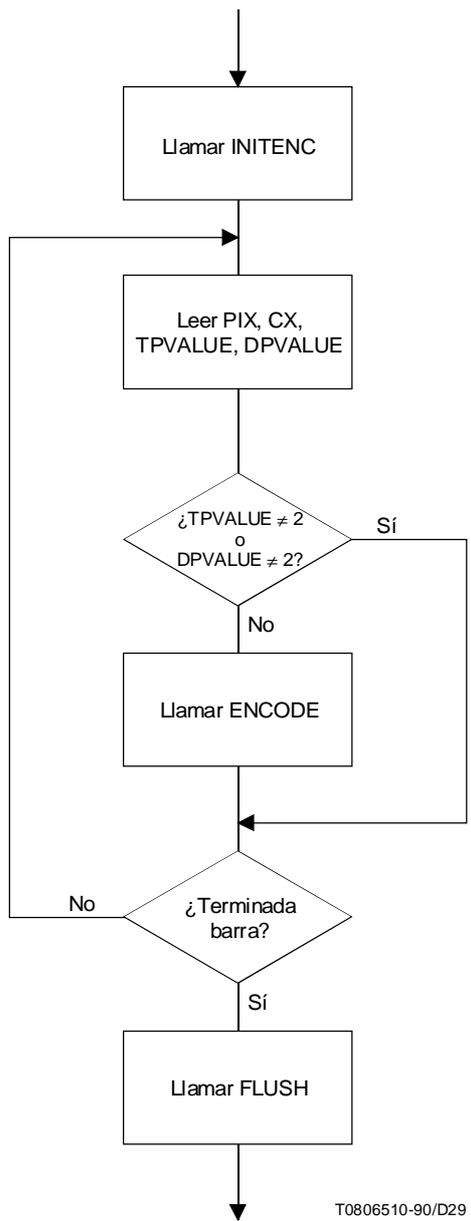
Con el proceso esbozado, la aproximación en el proceso de subdivisión de intervalos puede a veces hacer que el subintervalo LPS sea mayor que el subintervalo de MPS. Si, por ejemplo el valor de LSZ es $0,33 \times 0x10000$ y A es el mínimo valor admitido de $0x8000$, la proporcionalización aproximada da $1/3$ de intervalo al MPS y $2/3$ al LPS. Para evitar esta inversión de tamaño, el intervalo se subdivide utilizando esta sencilla aproximación, pero las asignaciones de intervalos de MPS y LPS se intercambian siempre que el intervalo de LPS sea mayor que el intervalo de MPS. Este «intercambio condicional» de MPS/LPS puede sólo producirse cuando sea necesaria una renormalización.

Siempre que se produce una renormalización, se invoca el proceso de estimación de probabilidad que determina una nueva estimación de probabilidad para el contexto que se codifica en ese momento.

6.8.2 Codificador

6.8.2.1 Diagrama de flujo del codificador

Este diagrama de flujo se ejecuta para cada barra de cada capa de resolución. Los pixels que no son típicamente predecibles ni predecibles de modo determinista se codifican por el procedimiento ENCODE. El procedimiento de inicialización INITENC se llama a la entrada, y el procedimiento de terminación FLUSH a la salida (véase la Figura 21).



T0806510-90/D29

Figura 21 – Diagrama de flujo del codificador

6.8.2.2 Convenios de registro de código del codificador

Los diagramas de flujo indicados en este punto suponen la estructura de registro presentada en el Cuadro 23.

Cuadro 23 – Estructura de registro del codificador

	msb		lsb	
Registro C	0000cbbb,	bbbbssss,	xxxxxxx,	xxxxxxx
Registro A	00000000,	0000000a,	aaaaaaaa,	aaaaaaaa

Los bits «a» son los bits fraccionarios en el valor de intervalo vigente y los bits «x» son los bits fraccionarios en el registro de código. Los bits «s» son bits espaciadores, de los cuales uno al menos es necesario para forzar el arrastre, y los bits «b» indican las posiciones de bits de las cuales los bytes de datos completados son suprimidos del registro C. El bit «c» es un bit de arrastre. El decimoséptimo registro A está conceptualmente presente, por lo que se muestra aquí, pero puede fácilmente evitarse si se desea una realización de 16 bits. En este caso, la inicialización a 0x0000 en lugar de 0x10000 opera correctamente mientras la infrautilización del soporte físico o lógico subyacente produzca los mismos 16 bits de bajo orden al sustraer de 0x0000 que al sustraer de 0x10000. Este comportamiento es el usual.

Estos convenios de registro ilustran una posible realización. Aquí especialmente existen muchas otras posibilidades.

6.8.2.3 Tablas de estimación de probabilidad

Para cada posible valor del contexto CX hay almacenado un valor de un bit MPS [CX] y un valor de siete bits ST [CX], que juntos capturan completamente la estimación de probabilidad adaptativa asociada con ese contexto determinado. En el Cuadro 24 se muestran cuatro formaciones indizadas por ST [CX].

El color MPS es el color más probable (estimado) de PIX. LSZ es el tamaño de intervalo del LPS, que puede interpretarse como una probabilidad vía ecuación (8), aunque ninguna interpretación así necesita efectuarse ya que sólo LSZ intervienen en los cálculos posteriores.

Las formaciones NLPS y NMPS dan, respectivamente, el siguiente estado de estimación de probabilidad para una observación del LPS y el MPS. El desplazamiento dado por NMPS sólo se produce si además de observar el MPS, se produce una renormalización. Cuando se produce el desplazamiento dado por NLPS, habrá también una inversión de MPS [CX] si SWTCH [CX] es 1.

El Anexo D (informativo) explica por qué las entradas del cuadro 24 están como están.

6.8.2.4 Diagrama de flujo para el procedimiento ENCODE

Si el PIX del símbolo vigente es igual al valor que se estima más probable en ese momento, se llama a la rutina CODEMPS. En otro caso, se llama a CODELPS (véase la Figura 22).

6.8.2.5 Diagrama de flujo para el procedimiento CODELPS

El procedimiento CODELPS consta normalmente de la adición del subintervalo del MPS A-LSZ [ST[CX]] al tren de código y de la proporcionalización del intervalo al subintervalo LSZ [ST[CX]]. Va siempre seguido por una renormalización. Si SWTCH [ST[CX]] es 1, se invierte MPS [CX].

Sin embargo, en el caso de que el subintervalo de LPS sea mayor que el subintervalo de MPS, se produce el intercambio MPS/LPS condicional y se codifica el subsiguiente de MPS (véase la Figura 23).

6.8.2.6 Diagrama de flujo para el procedimiento CODEMPS

El procedimiento CODEMPS reduce normalmente el tamaño del intervalo al subintervalo de MPS. Sin embargo, si el subintervalo de LPS es mayor que el subintervalo de MPS, se produce el intercambio condicional y se codifica en su lugar el subintervalo de LPS. Obsérvese que esta inversión de tamaño de intervalo no puede producirse a menos que se requiera una renormalización después de la codificación del símbolo (véase la Figura 24).

Cuadro 24 – Tabla de estimación de probabilidad

ST	LSZ	NLPS	NMPS	SWTCH	ST	LSZ	NLPS	NMPS	SWTCH
0	0x5a1d	1	1	1	57	0x01a4	55	58	0
1	0x2586	14	2	0	58	0x0160	56	59	0
2	0x1114	16	3	0	59	0x0125	57	60	0
3	0x080b	18	4	0	60	0x00f6	58	61	0
4	0x03d8	20	5	0	61	0x00cb	59	62	0
5	0x01da	23	6	0	62	0x00ab	61	63	0
6	0x00e5	25	7	0	63	0x008f	61	32	0
7	0x006f	28	8	0	64	0x5b12	65	65	1
8	0x0036	30	9	0	65	0x4d04	80	66	0
9	0x001a	33	10	0	66	0x412c	81	67	0
10	0x000d	35	11	0	67	0x37d8	82	68	0
11	0x0006	9	12	0	68	0x2fe8	83	69	0
12	0x0003	10	13	0	69	0x293c	84	70	0
13	0x0001	12	13	0	70	0x2379	86	71	0
14	0x5a7f	15	15	1	71	0x1edf	87	72	0
15	0x3f25	36	16	0	72	0x1aa9	87	73	0
16	0x2cf2	38	17	0	73	0x174e	72	74	0
17	0x207c	39	18	0	74	0x1424	72	75	0
18	0x17b9	40	19	0	75	0x119c	74	76	0
19	0x1182	42	20	0	76	0x0f6b	74	77	0
20	0x0cef	43	21	0	77	0x0d51	75	78	0
21	0x09a1	45	22	0	78	0x0bb6	77	79	0
22	0x072f	46	23	0	79	0x0a40	77	48	0
23	0x055c	48	24	0	80	0x5832	80	81	1
24	0x0406	49	25	0	81	0x4d1c	88	82	0
25	0x0303	51	26	0	82	0x438e	89	83	0
26	0x0240	52	27	0	83	0x3bdd	90	84	0
27	0x01b1	54	28	0	84	0x34ee	91	85	0
28	0x0144	56	29	0	85	0x2eae	92	86	0
29	0x00f5	57	30	0	86	0x299a	93	87	0
30	0x00b7	59	31	0	87	0x2516	86	71	0
31	0x008a	60	32	0	88	0x5570	88	89	1
32	0x0068	62	33	0	89	0x4ca9	95	90	0
33	0x004e	63	34	0	90	0x44d9	96	91	0
34	0x003b	32	35	0	91	0x3e22	97	92	0
35	0x002c	33	9	0	92	0x3824	99	93	0
36	0x5ae1	37	37	1	93	0x32b4	99	94	0
37	0x484c	64	38	0	94	0x2e17	93	86	0
38	0x3a0d	65	39	0	95	0x56a8	95	96	1
39	0x2ef1	67	40	0	96	0x4f46	101	97	0
40	0x261f	68	41	0	97	0x47e5	102	98	0
41	0x1f33	69	42	0	98	0x41cf	103	99	0
42	0x19a8	70	43	0	99	0x3c3d	104	100	0
43	0x1518	72	44	0	100	0x375e	99	93	0
44	0x1177	73	45	0	101	0x5231	105	102	0
45	0x0e74	74	46	0	102	0x4c0f	106	103	0
46	0x0bfb	75	47	0	103	0x4639	107	104	0
47	0x09f8	77	48	0	104	0x415e	103	99	0
48	0x0861	78	49	0	105	0x5627	105	106	1
49	0x0706	79	50	0	106	0x50e7	108	107	0
50	0x05cd	48	51	0	107	0x4b85	109	103	0
51	0x04de	50	52	0	108	0x5597	110	109	0
52	0x040f	50	53	0	109	0x504f	111	107	0
53	0x0363	51	54	0	110	0x5a10	110	111	1
54	0x02d4	52	55	0	111	0x5522	112	109	0
55	0x025c	53	56	0	112	0x59eb	112	111	1
56	0x01f8	54	57	0					

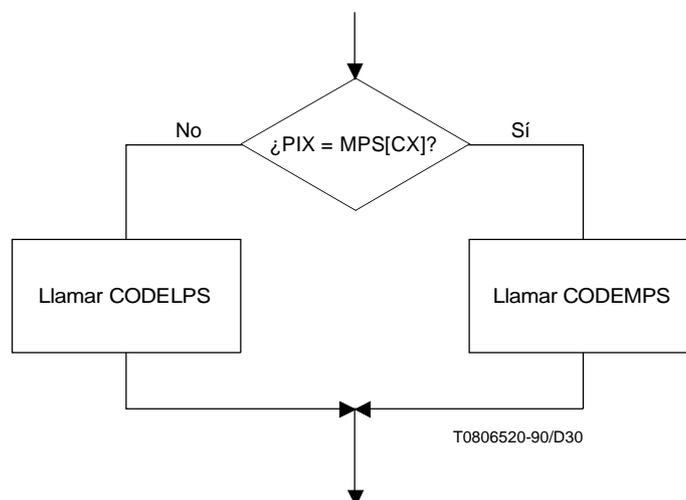


Figura 22 – Diagrama de flujo para el procedimiento ENCODE

6.8.2.7 Diagrama de flujo para el procedimiento RENORME

Tanto el registro de intervalo A como el registro de código C se desplazan, un bit cada vez. Se cuenta el número de desplazamientos en el contador CT , y cuando se cuenta CT hasta cero, se suprime de C un byte de datos comprimidos por el procedimiento `BYTEOUT`. La renormalización continúa hasta que A deja de ser menor que $0x8000$ (véase la Figura 25).

6.8.2.8 Diagrama de flujo para el procedimiento BYTEOUT

El procedimiento `BYTEOUT` es llamado desde `RENORME`. La variable $TEMP$ es una variable temporal que mantiene el byte en la parte superior del registro C que debe hacerse salir más una indicación de transporte. La variable $BUFFER$ mantiene la salida provisional más reciente distinta de $0xff$. El contador SC mantiene el número de bytes $0xff$ que ha habido desde que se sacó provisionalmente el byte de $BUFFER$ (véase la Figura 26).

El desplazamiento del registro de código en 19 bits alinea los bits de salida «b» con los bits de bajo orden de $TEMP$. La primera prueba determina entonces si se ha producido un arrastre. Si así ha sido, el transporte debe añadirse al byte la salida provisional de $BUFFER$, porque finalmente está comprometido a salir. Cualesquiera bytes de salida apilados (convertidos a cero por el transporte) se sacan a continuación. Finalmente, el nuevo byte provisional de salida $BUFFER$ se pone igual a $TEMP$ menos cualquier transporte.

Si no se ha producido ningún transporte, el byte de salida se comprueba para ver si es $0xff$. Si así es, se aumenta la cuenta de pila SC , y la salida debe demorarse hasta que se resuelva el transporte. De no ser así, se ha resuelto el transporte, y pueden entonces sacarse cualesquiera bytes $0xff$ apilados.

NOTA – La probabilidad de que el contador SC alcance un determinado número n decrece rápidamente siguiendo una ley 2^{-8n} , de modo que en la práctica los valores de SC superiores a 3 ó 4 son raramente vistos al codificar una imagen. Sin embargo, en principio SC puede resultar tan grande como el número de bytes en el fichero de salida SCD . Siempre que se resuelva el transporte, la imagen de entrada no puede procesarse mientras se producen bytes SC $0x00$ ó $0xff$. Dado que SC puede resultar en principio bastante grande, esta detención puede también en principio resultar bastante larga.

Si es importante para una determinada realización, el marcador reservado puede utilizarse para limitar finitamente cualquier detención del procesamiento de la imagen de entrada. Una manera es insertar el marcador reservado siempre que SC alcance algún número pequeño, por ejemplo 8, y luego decrementar SC en 8. Entonces, en un paso posprocesamiento dentro del codificador, cada uno de estos marcadores es sustituido por ocho bytes $0xff$ u ocho bytes $0x00$, según convenga. Este posprocesamiento debe efectuarlo el decodificador, ya que ningún decodificador se espera que conozca nada sobre dicha aplicación del marcador reservado.

Un segundo modo de utilizar el marcador reservado para este mismo fin es permitir que SC se eleve a valores arbitrariamente grandes, pero si SC es mayor que algún número, por ejemplo 8, cuando el transporte está eventualmente resuelto, producir el marcador reservado seguido por un byte $0x00$ ó $0xff$ para indicar a qué transporte se resolvió y luego una codificación en, por ejemplo, los cuatro próximos bytes, del valor SC real. Una vez más, se requiere posprocesamiento por el codificador para sustituir este marcador por el número adecuado de bytes $0xff$ ó $0x00$.

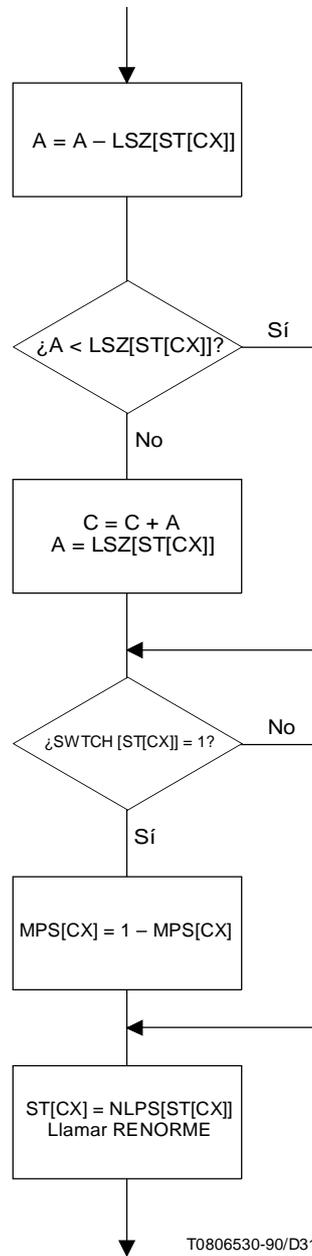


Figura 23 – Diagrama de flujo para el procedimiento CODELPS

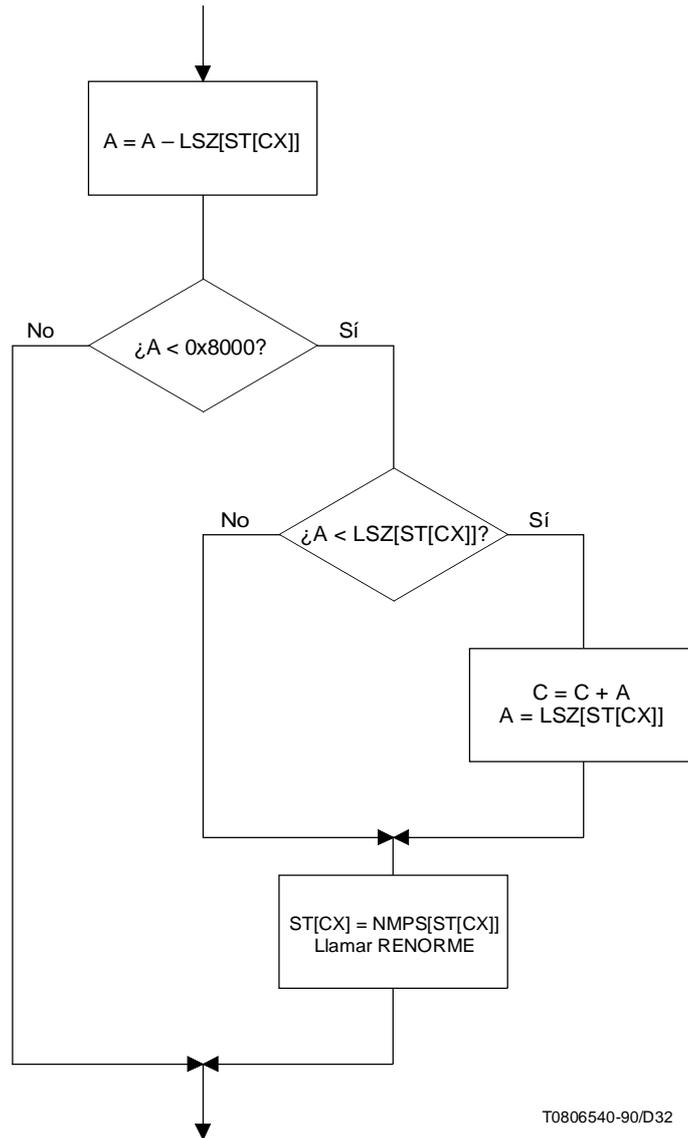


Figura 24 – Diagrama de flujo para el procedimiento CODEMPS

6.8.2.9 Diagrama de flujo para el procedimiento INITENC

Si esta barra está en la parte superior de la imagen, los estados de estimación de probabilidad para todos los posibles valores de CX se ponen a 0 (es decir, el estado equiprobable). En otro caso, se reponen a sus valores al final de la última barra en esta resolución. Se libera la cuenta de pila SC y el registro de código C . El contador CT se pone a 11 (un byte más los 3 bits espaciadores). El registro de intervalo de codificación A se pone a $0x10000$. Alternativamente, para una realización de 16 bits, puede ponerse a $0x0000$ mientras que el soporte lógico o el soporte físico produzca los mismos 16 bits al sustraer de 0 una magnitud de 16 bits que la que se obtiene matemáticamente sustrayendo de $0x10000$. Así ocurrirá siempre (véase la Figura 27).

6.8.2.10 Diagrama de flujo para el procedimiento FLUSH

Se llaman primero dos subprocedimientos. Se suprime luego el primer byte que se escribió en el tren SCD , y si se desean alguno o todos de cualesquiera bytes $0x00$ al final de SCD se eliminan también hasta que finalmente aparezca un byte distinto de $0x00$. Las buenas realizaciones de soporte lógico y soporte físico constituirán variables auxiliares para que estos bytes nunca se escriban en el mismo lugar. La realización aquí descrita se eligió por su sencillez y concisión (véase la Figura 28).

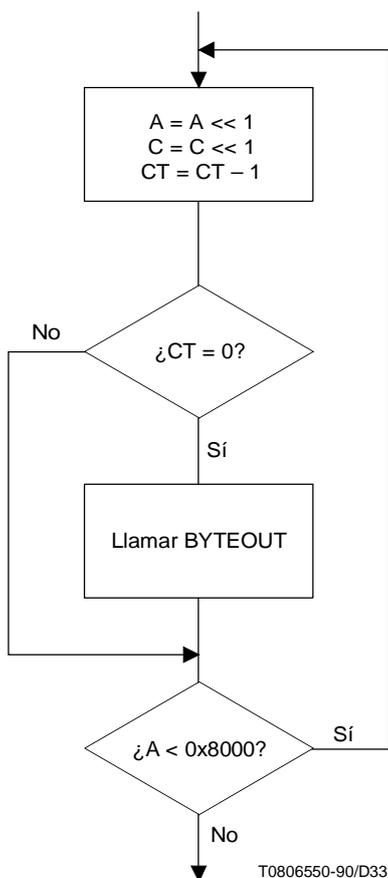


Figura 25 – Diagrama de flujo para el procedimiento RENORME

6.8.2.11 Diagrama de flujo para el procedimiento CLEARBITS

El registro de código *C* se pone al valor de [*C*, *C*+*A*-1] que termina con el mayor número posible de bits cero (véase la Figura 29).

6.8.2.12 Diagrama de flujo para el procedimiento FINALWRITES

La resolución de transporte final se realiza y se escriben dos bytes procedentes de *C* (véase la Figura 30).

6.8.3 Decodificador

6.8.3.1 Diagrama de flujo del decodificador

Este diagrama de flujo se ejecuta para cada barra de cada capa de resolución. Los pixels que no son típicamente predecibles ni determinísticamente predecibles se decodifican por el procedimiento DECODE. El procedimiento de inicialización INITDEC se llama a la entrada (véase la Figura 31).

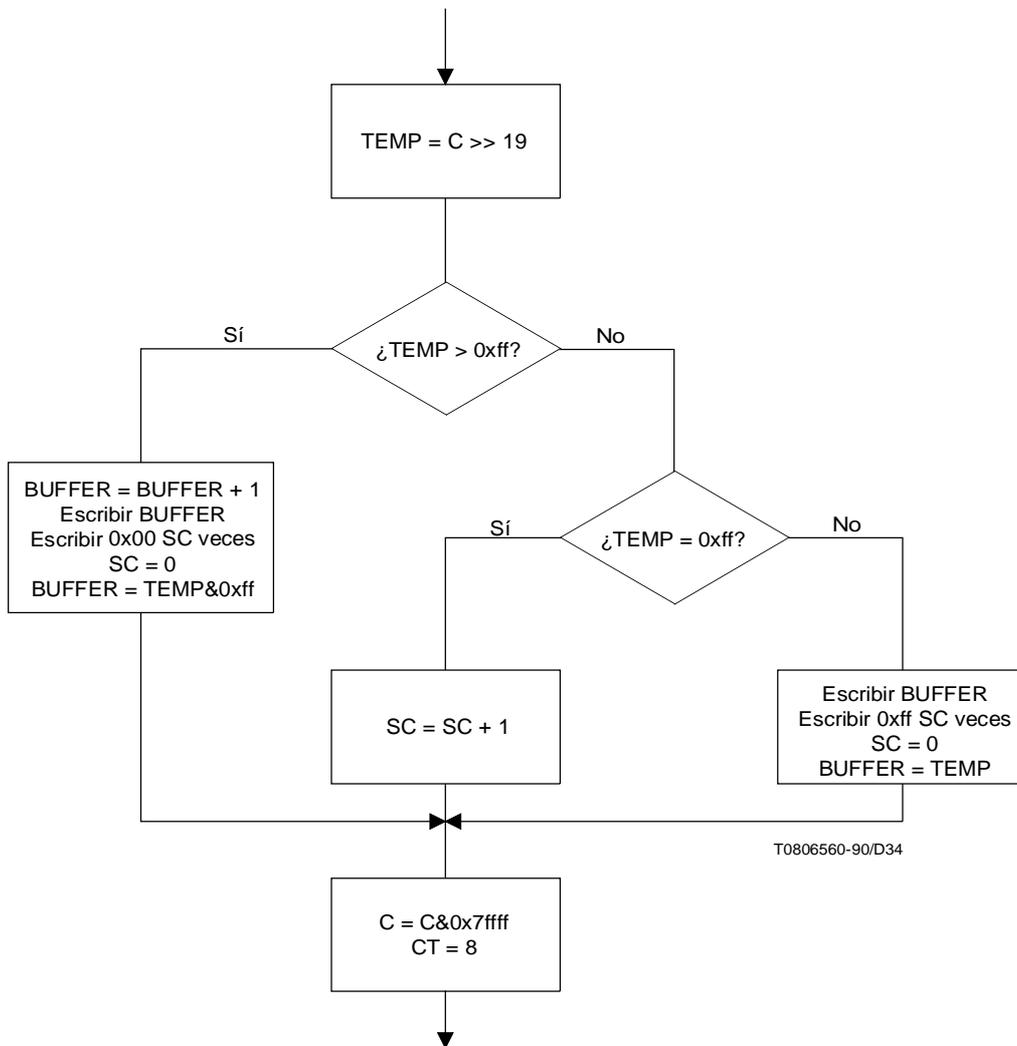


Figura 26 – Diagrama de flujo para el procedimiento BYTEOUT

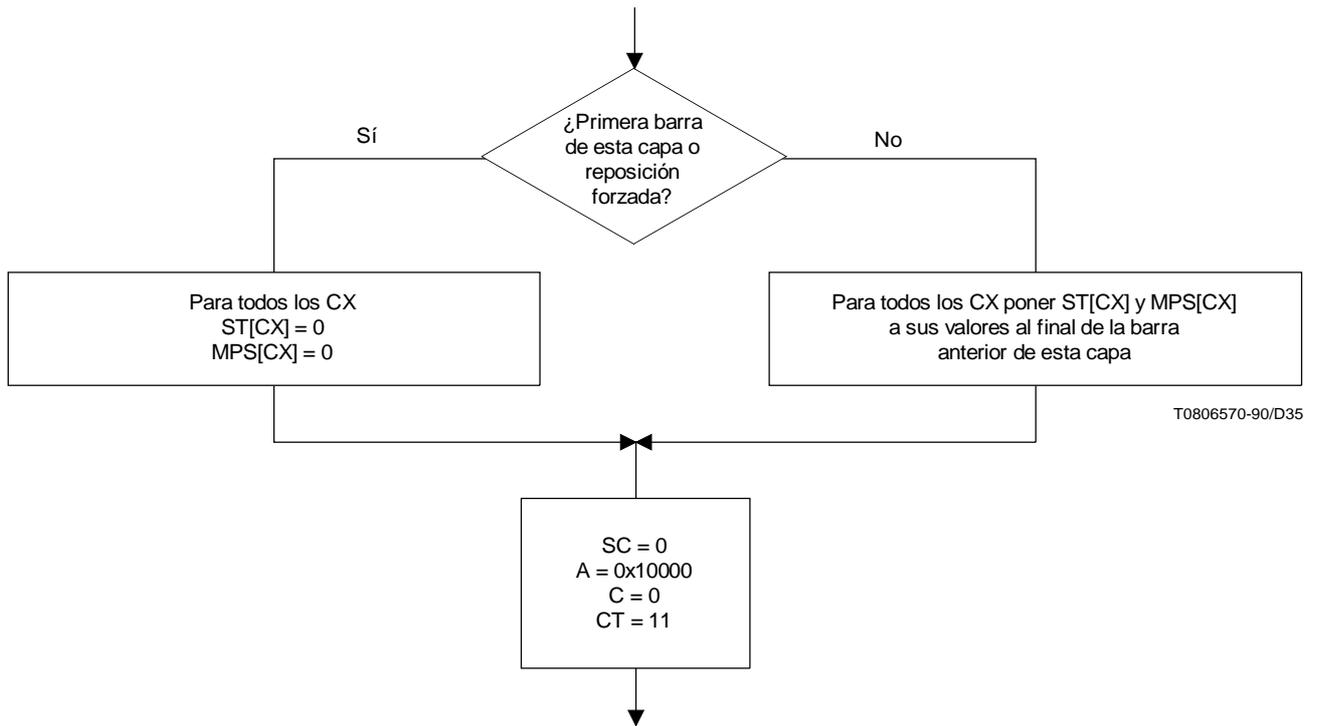


Figura 27 – Diagrama de flujo para el procedimiento INITENC

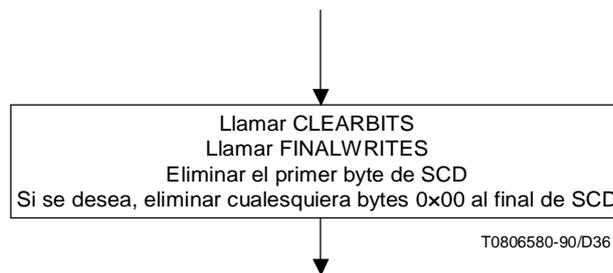


Figura 28 – Diagrama de flujo para el procedimiento FLUSH

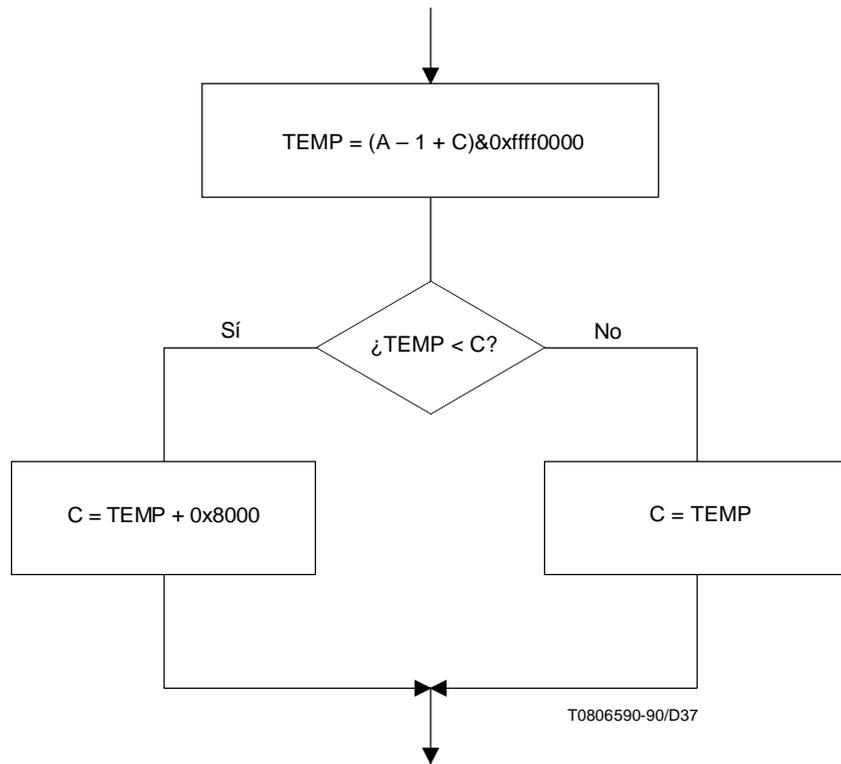


Figura 29 – Diagrama de flujo para el procedimiento CLEARBITS

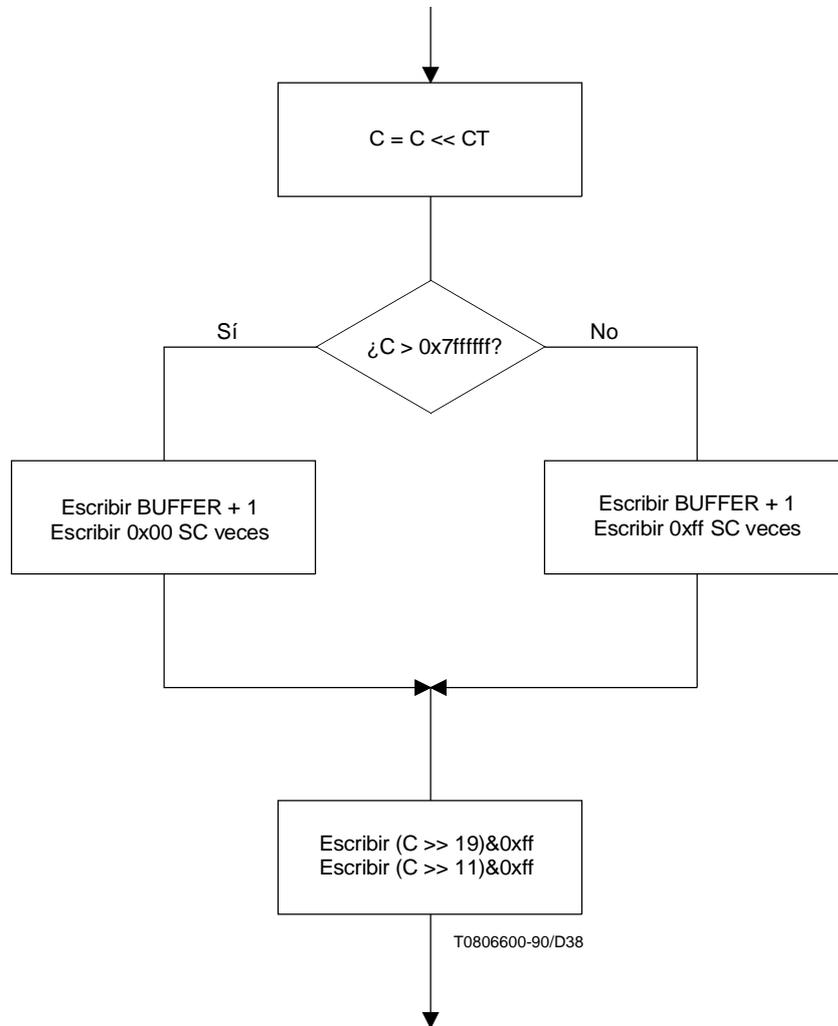


Figura 30 – Diagrama de flujo para el procedimiento FINALWRITES

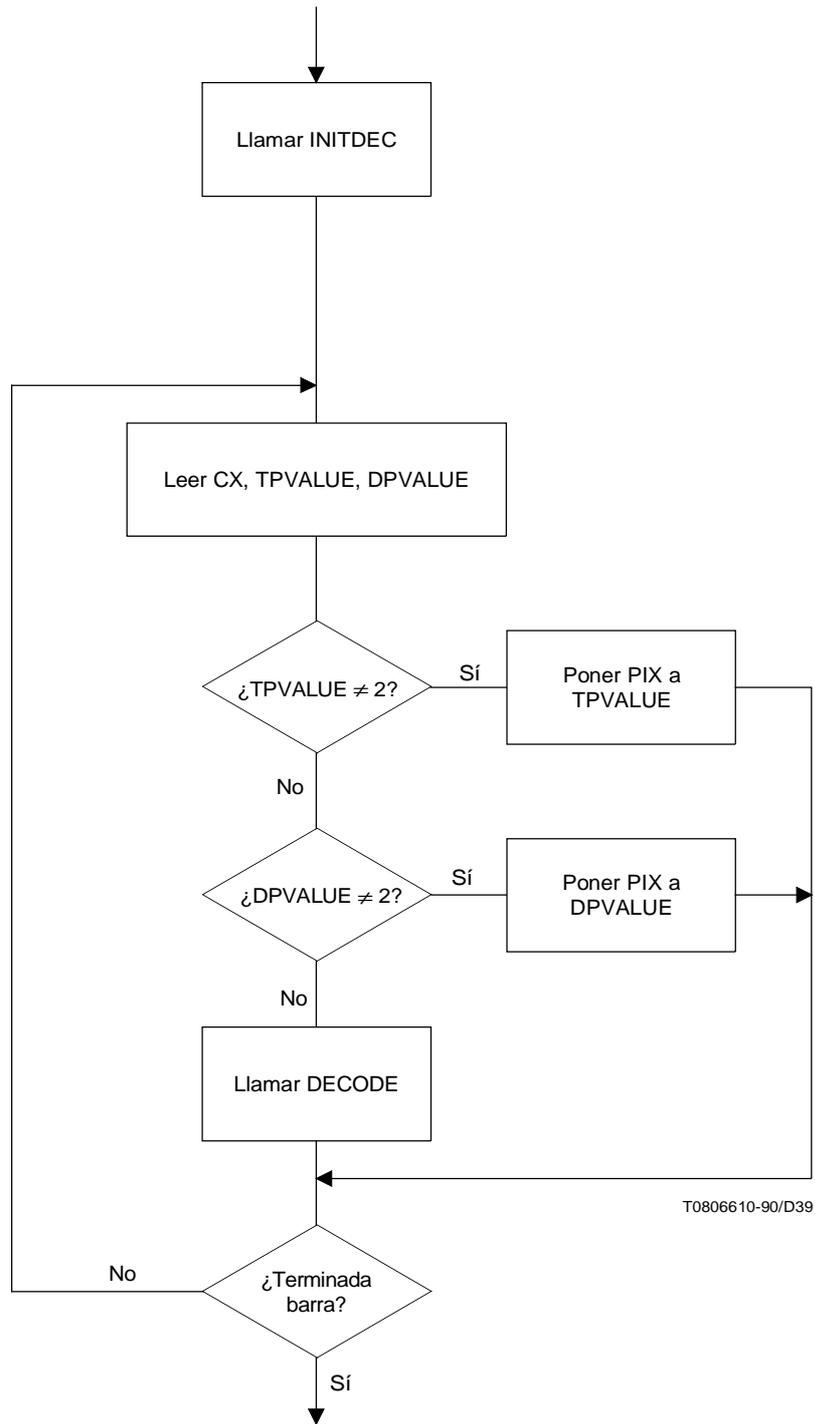


Figura 31 – Diagrama de flujo del decodificador

6.8.3.2 Convenios de registro de código del decodificador

Los diagramas de flujo indicados en este punto suponen la estructura de registro presentada en el Cuadro 25.

Cuadro 25 – Estructura de registro del decodificador

	msb	lsb
Registro CHIGH	xxxxxxx,	xxxxxxx
Registro CLOW	bbbbbbb,	0000000
Registro A	a, aaaaaaa,	aaaaaaa

CHIGH y CLOW pueden considerarse como un registro C de 32 bits, en el que la renormalización de C desplaza un bit de nuevos datos del bit 15 de CLOW al bit 0 de CHIGH. Sin embargo, las comparaciones de decodificación utilizan CHIGH solamente. Los nuevos datos se insertan en los bits «b» de CLOW un byte cada vez. Como el decodificador, el decimoséptimo bit del registro A está conceptualmente presente, pero es fácilmente evitado en las realizaciones.

6.8.3.3 Tablas de estimación de probabilidad

Las tablas de estimación de probabilidad utilizadas en la decodificación son idénticas a las utilizadas en la codificación.

6.8.3.4 Diagrama de flujo para el procedimiento DECODE

Sólo cuando se necesita un renormalización, es posible que pueda haberse producido el intercambio condicional MPS/LPS (véase la Figura 32).

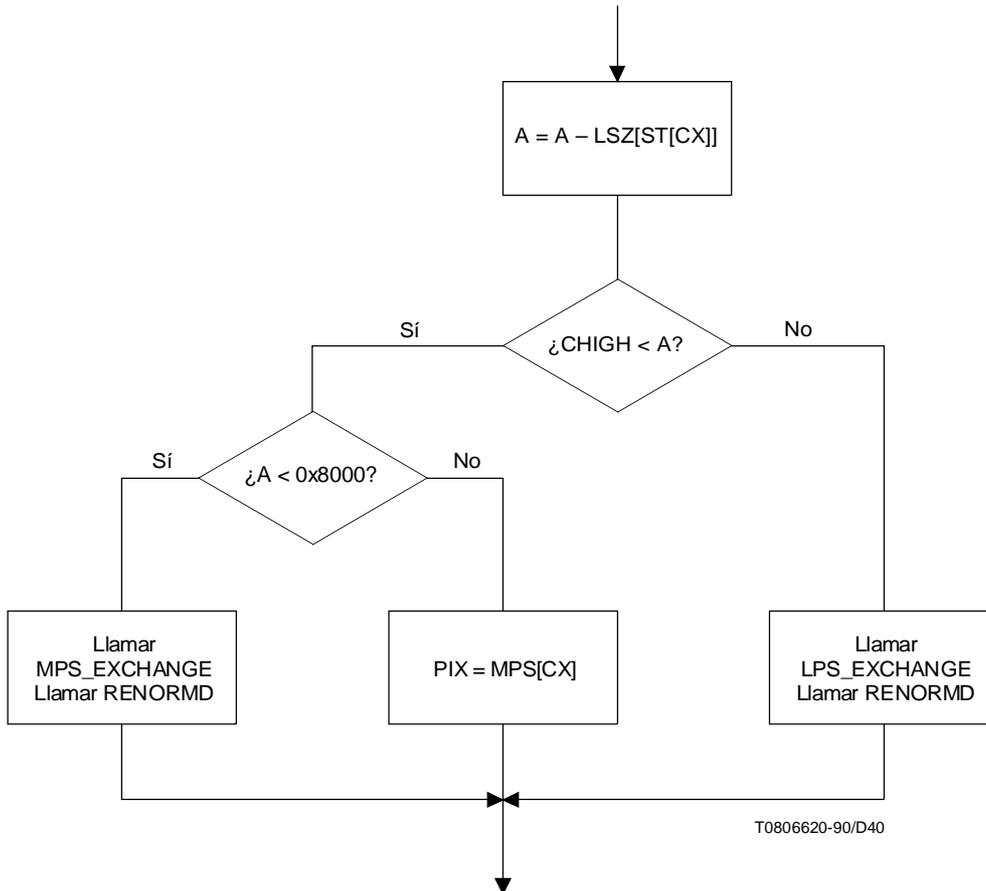


Figura 32 – Diagrama de flujo para el procedimiento DECODE

6.8.3.5 Diagrama de flujo para el procedimiento LPS-EXCHANGE

Véase la Figura 33.

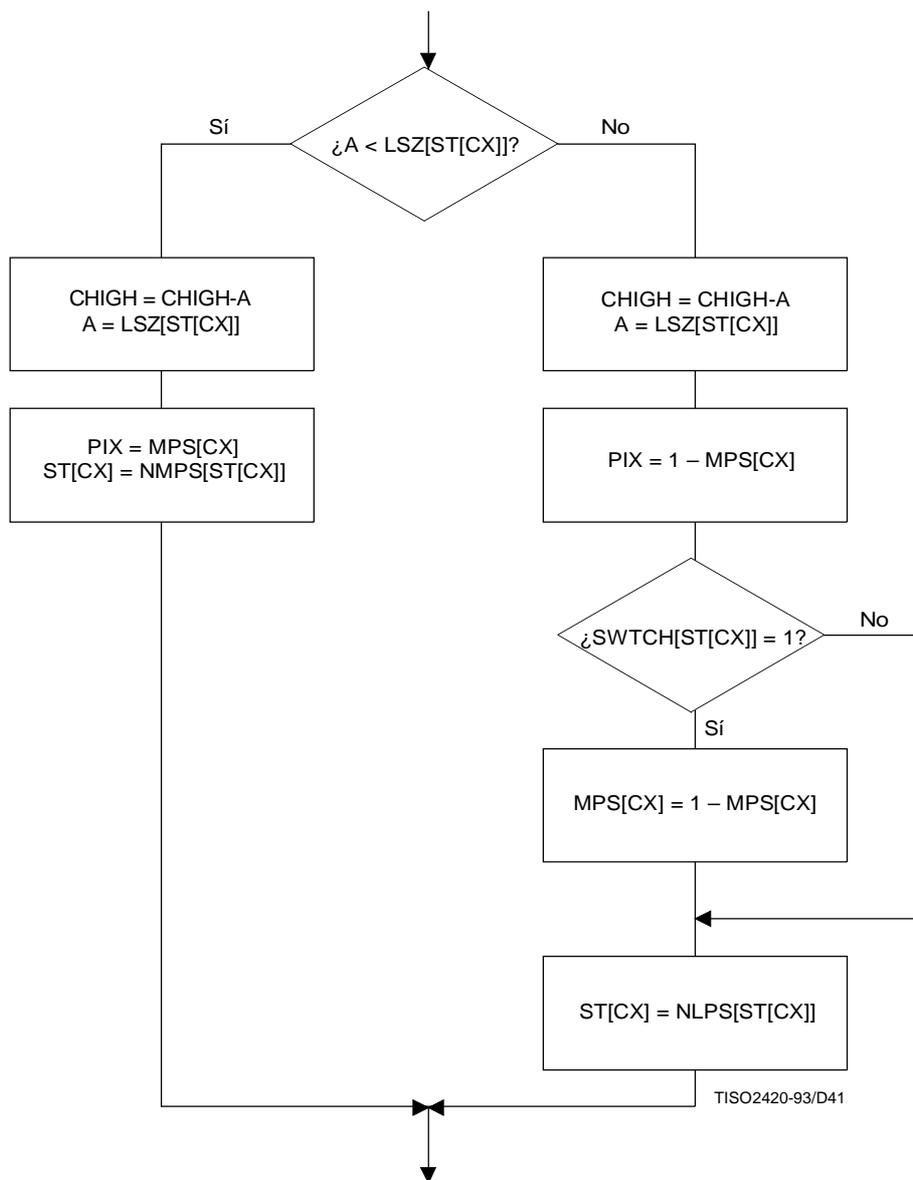


Figura 33 – Diagrama de flujo para el procedimiento LPS_EXCHANGE

6.8.3.6 Diagrama de flujo para el procedimiento MPS-EXCHANGE

Véase la Figura 34.

6.8.3.7 Diagrama de flujo para el procedimiento RENORMD

CT es un contador que controla el número de bits comprimidos en la sección CLOW del registro C. Cuando CT es cero, se inserta un nuevo byte en CLOW.

Tanto el registro de intervalo A como el registro de código C se desplazan, un bit cada vez, hasta que A deja de ser menor que 0x8000 (véase la Figura 35).

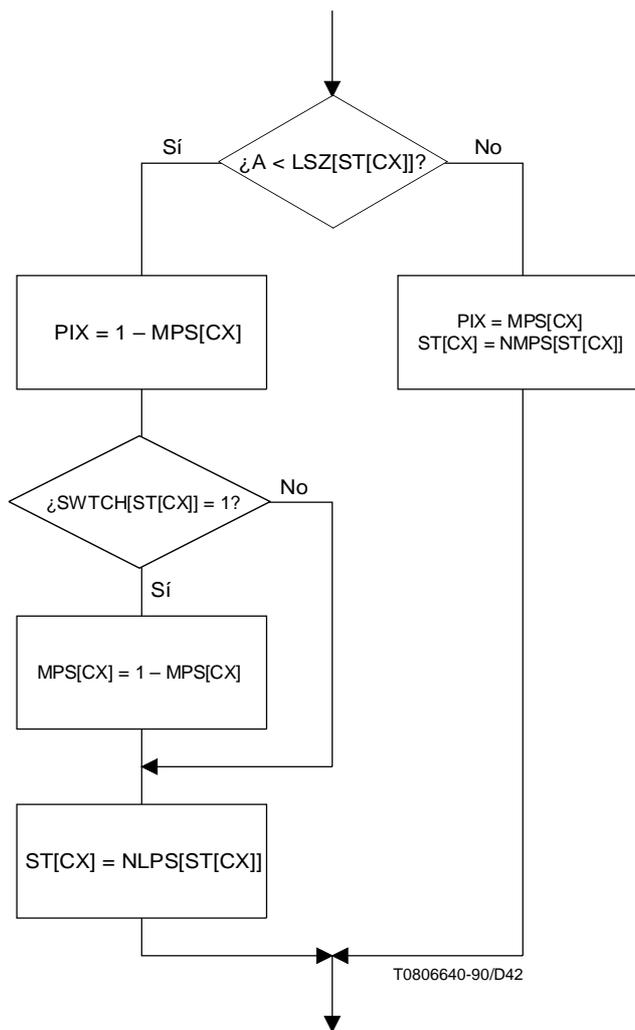


Figura 34 – Diagrama de flujo para el procedimiento MPS_EXCHANGE

6.8.3.8 Diagrama de flujo para el procedimiento BYTEIN

Los bytes se leen de SCD hasta que se agota, tras lo cual las lecturas posteriores se satisfacen devolviendo 0x00. Los bytes leídos se insertan en los 8 bits superiores de CLOW. El contador se repone a 8 (véase la Figura 36).

6.8.3.9 Diagrama de flujo para el procedimiento INITDEC

Si esta barra está en la parte superior de la imagen, los estados de estimación de probabilidad para todos los posibles valores de CX se ponen a 0. En otro caso, se repone a sus valores al final de la última barra en esta resolución. Se leen tres bytes en el registro C (véase la Figura 37).

7 Métodos de prueba y ejemplos de trenes de datos

Esta cláusula normativa describe métodos de prueba para el algoritmo descrito en cláusulas anteriores de esta Especificación. Existen muchas parametrizaciones posibles, y este punto explicará algunos métodos para probar la exactitud de algunas de ellas, consideradas de utilidad para depurar las realizaciones.

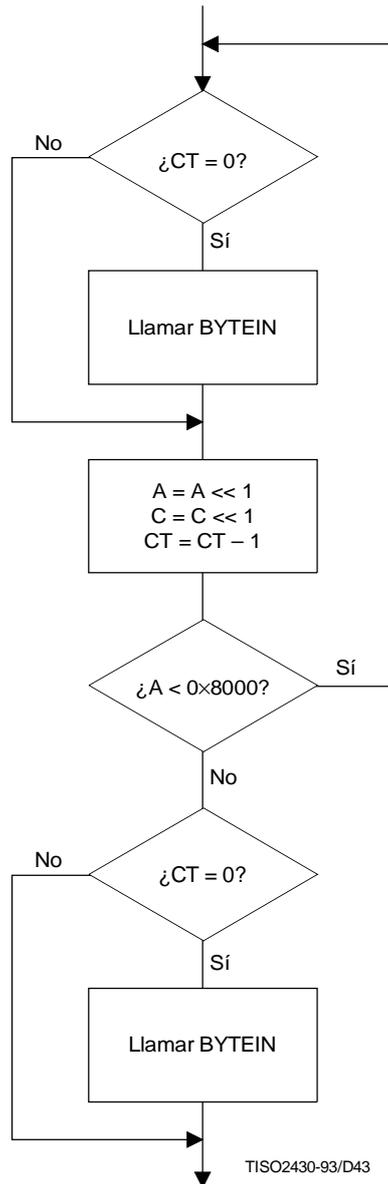


Figura 35 – Diagrama de flujo para el procedimiento RENORMD

7.1 Codificación aritmética

En esta subcláusula se proporciona un pequeño conjunto de datos para probar el codificador y decodificador aritméticos. Se supondrá que este conjunto de datos representa los datos en bruto de una barra en orden de barrido por puntos del MSB al LSB. La prueba se estructura para probar muchos de los trayectos del codificador y decodificador, pero es imposible en una corta secuencia de prueba comprobar todos ellos, por lo que desgraciadamente la concordancia con los resultados de esta prueba no garantiza completamente la correcta realización.

PIX: 05e0 0000 8b00 01c4 1700 0034 7fff 1a3f 951b 05d8 1d17 e770 0000 0000 0656 0e6a

CX: 0fe0 0000 0f00 00f0 ff00 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

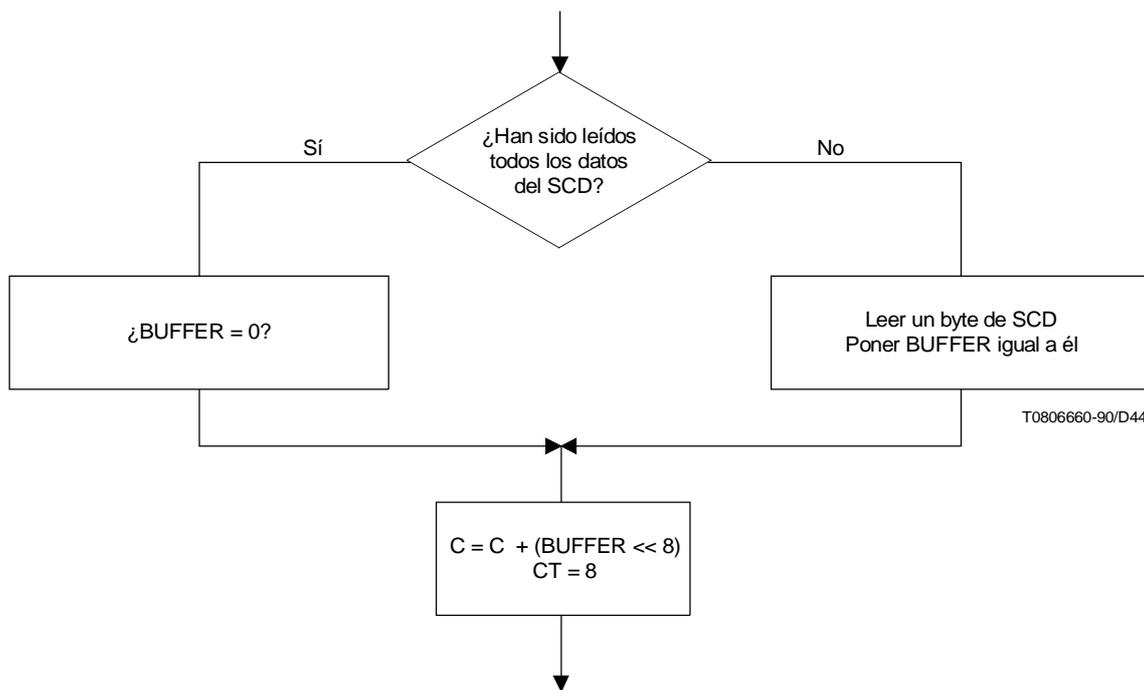


Figura 36 – Diagrama de flujo para el procedimiento BYTEIN

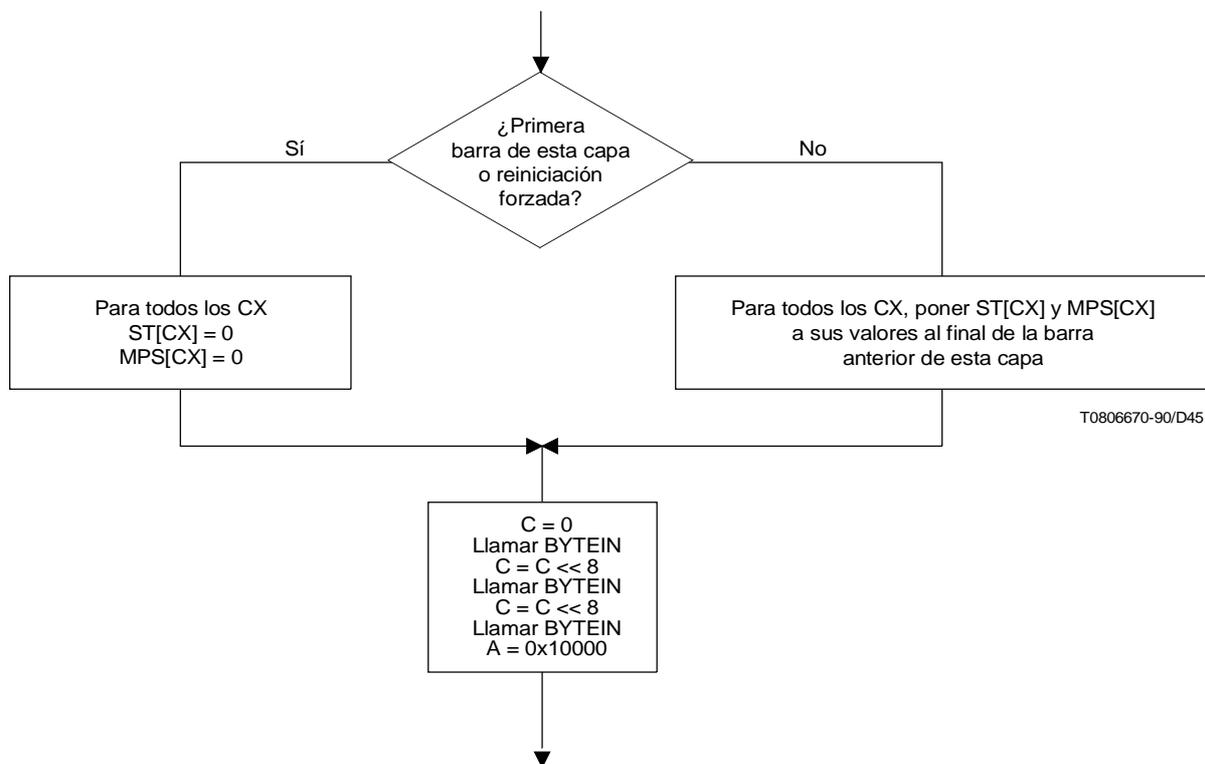


Figura 37 – Diagrama de flujo para el procedimiento INITDEC

El codificador (véase la Figura 18) tiene cuatro entradas. La primera secuencia describe la entrada PIX y la segunda la correspondiente entrada CX. Por sencillez, la variable de entrada CX de esta prueba asume sólo los dos valores 0 y 1, y no los 4096 valores que asume para el codificador/decodificador JBIG. Las otras dos entradas TPVALUE y DPVALUE se supone que son siempre 2 (para el codificador y el decodificador). La salida del codificador (SCD) consta de 200 bits (25 bytes) y se muestra a continuación en forma hex:

```
SCD: 6989 995c 32ea faa0 d5ff 527f ffff ffc0 0000 003f ff2d 2082 91
```

Para el decodificador (véase la Figura 19), las entradas SCD y CX se dan a partir de las secuencias anteriores, mientras que PIX es ahora la salida.

El Cuadro 26 proporciona una lista símbolo a símbolo de la operación del codificador y decodificador aritméticos. La primera línea de esta tabla corresponde a los procedimientos INITENC e INITDEC con BUFFER inicializado a 0x00, y la última corresponde al procedimiento FINALWRITES del codificador. La primera columna es el contador de eventos (EC), la segunda el valor del evento binario PIX a codificar y decodificar, y la tercera el valor correspondiente de la variable de entrada CX. La columna MPS indica el sentido de MPS, y la columna CE indica que el intercambio condicional (véanse 6.8.2.5, 6.8.2.6 y 6.8.3.4) se producirá cuando se codifique (decodifique) el evento binario vigente PIX. El estado vigente (véase el Cuadro 24) y su correspondiente tamaño LPS se muestran en las columnas designadas por ST y LSZ. A continuación se indica el valor del registro A antes de que se codifique (decodifique) el evento. Obsérvese que el registro A es siempre mayor o igual que 0x8000.

Las variables hasta este punto eran comunes para el codificador y el decodificador. Las cinco columnas siguientes (C, CT, SC, BUF, OUT) son sólo para el decodificador, y las tres últimas columnas (C, CT, IN) son para el decodificador. Para el codificador las entradas son (PIX, CX) y sus salidas se dan en la columna OUT, mientras que para el decodificador las entradas son (CX, IN) y la salida es PIX. Los valores del registro C indicados en la columna C se dan antes de que se codifique (decodifique) el evento vigente. Para el codificador, CT es un contador que indica cuándo está listo un byte para la salida del registro C. SC es el número de bytes 0xff apilados en el codificador en espera de la resolución del arrastre. La columna BUF muestra el byte BUFFER variable en espera de ser enviado. Este byte puede a veces cambiar a partir de un arrastre. Finalmente, para el codificador los bytes de código se indican en la columna OUT. Estos bytes se envían a la salida durante la codificación del evento vigente. Si se consigna más de un byte, estos bytes se sacaron también durante el evento vigente y se generaron liberando el contador SC.

Para el decodificador, los valores del registro C se indican antes de que el evento se decodifique, y se consignan en la columna C. El contador del decodificador CT se muestra en la próxima columna, e indica cuándo introducir el próximo byte procedente del tren de código. Finalmente, en la última columna, los bytes de código se consignan si se leyeron en el registro de código al final del evento vigente.

La última fila muestra la salida generada por el procedimiento FINALWRITES. Este procedimiento genera también cinco bytes 0x00 adicionales, dos de ellos procedentes de la liberación del computador y tres del vaciado del registro C. Estos bytes 0x00 finales no se incluyen en los bytes de código SCD, ya que se ejerció la opción de eliminar todos los bytes 0x00 del final del tren de código de una barra (véase el procedimiento FLUSH en la Figura 28). Obsérvese que es admisible dejar cualquiera de esos bytes 0x00 al final del tren de código SCD. El decodificador ahora, al llegar al fin del tren codificado, lee los 0x00 en su registro C hasta que decodifica el número deseado (256) de pixels.

A fin de generar ahora el tren de código PSCD (véase el Cuadro 12), debe insertarse el byte STUFF (0x00) después de cada byte ESC (0xff) en SCD.

```
PSCD: 6989 995c 32ea faa0 d5ff 0052 7fff 00ff 00ff 00c0 0000 003f ff00 2d20 8291
```

Finalmente, debido a la suposición de que los datos (CX, PIX) son los datos en bruto de una barra, la entidad de datos de barra (SDE) puede ser generada agregando los bytes ESC (0xff) y SDNORM (0x02) (no hay ningún ATMOVE para esta barra).

```
SDE: 6989 995c 32ea faa0 d5ff 0052 7fff 00ff 00ff 00c0 0000 003f ff00 2d20 8291 ff02
```

Cuadro 26 – Datos de seguimiento del codificador y decodificador

EC	PIX	CX	MPS	CE	ST	LSZ hex	A hex	CODIFICADOR				DECODIFICADOR			
								C hex	CT	CS	BUF hex	OUT hex	C hex	CT	IN hex
0													00000000	0	698999
1	0	0	0	0	0	5a1d	10000	00000000	11	0	00		69899900	8	
2	0	0	0	1	0	5a1d	0a5e3	00000000	11	0	00		69899900	8	
3	0	0	0	0	1	2586	0b43a	0000978c	10	0	00		3b873200	7	
4	0	0	0	0	1	2586	08eb4	0000978c	10	0	00		3b873200	7	
5	0	1	0	0	0	5a1d	0d25c	00012f18	9	0	00		770e6400	6	
6	1	1	0	0	1	2586	0f07e	00025e30	8	0	00		ee1cc800	5	
7	0	1	0	1	14	5a7f	09618	000ca4a0	6	0	00		8c932000	3	
8	1	1	0	0	15	3f25	0b4fe	0019c072	5	0	00		a1f44000	2	5c
9	1	1	0	0	36	5ae1	0fc94	0068d92c	3	0	00		b06d5c00	8	
10	1	1	1	0	37	484c	0b5c2	00d2f5be	2	0	00		1d74b800	7	
11	1	1	1	0	38	3a0d	0daec	01a5eb7c	1	0	00		3ae97000	6	
12	0	0	0	0	2	1114	0a0df	01a5eb7c	1	0	00		3ae97000	6	
13	0	0	0	0	2	1114	08fcb	01a5eb7c	1	0	00		3ae97000	6	
14	0	0	0	0	3	080b	0fd6e	0003d6f8	8	0	69		75d2e000	5	
15	0	0	0	0	3	080b	0f563	0003d6f8	8	0	69		75d2e000	5	
16	0	0	0	0	3	080b	0ed58	0003d6f8	8	0	69		75d2e000	5	
17	0	0	0	0	3	080b	0e54d	0003d6f8	8	0	69		75d2e000	5	
18	0	0	0	0	3	080b	0dd42	0003d6f8	8	0	69		75d2e000	5	
19	0	0	0	0	3	080b	0d537	0003d6f8	8	0	69		75d2e000	5	
20	0	0	0	0	3	080b	0cd2c	0003d6f8	8	0	69		75d2e000	5	
21	0	0	0	0	3	080b	0c521	0003d6f8	8	0	69		75d2e000	5	
22	0	0	0	0	3	080b	0bd16	0003d6f8	8	0	69		75d2e000	5	
23	0	0	0	0	3	080b	0b50b	0003d6f8	8	0	69		75d2e000	5	
24	0	0	0	0	3	080b	0ad00	0003d6f8	8	0	69		75d2e000	5	
25	0	0	0	0	3	080b	0a4f5	0003d6f8	8	0	69		75d2e000	5	
26	0	0	0	0	3	080b	09cea	0003d6f8	8	0	69		75d2e000	5	
27	0	0	0	0	3	080b	094df	0003d6f8	8	0	69		75d2e000	5	
28	0	0	0	0	3	080b	08cd4	0003d6f8	8	0	69		75d2e000	5	
29	0	0	0	0	3	080b	084c9	0003d6f8	8	0	69		75d2e000	5	
30	0	0	0	0	4	03d8	0f97c	0007adf0	7	0	69		eba5c000	4	
31	0	0	0	0	4	03d8	0f5a4	0007adf0	7	0	69		eba5c000	4	
32	0	0	0	0	4	03d8	0f1cc	0007adf0	7	0	69		eba5c000	4	
33	1	0	0	0	4	03d8	0edf4	0007adf0	7	0	69		eba5c000	4	32
34	0	0	0	0	20	0cef	0f600	02260300	1	0	69		6270c800	6	
35	0	0	0	0	20	0cef	0e911	02260300	1	0	69		6270c800	6	
36	0	0	0	0	20	0cef	0dc22	02260300	1	0	69		6270c800	6	
37	1	1	1	0	38	3a0d	0cf33	02260300	1	0	69		6270c800	6	
38	0	1	1	0	38	3a0d	09526	02260300	1	0	69	69	6270c800	6	
39	1	1	1	0	65	4d04	0e834	00097864	7	0	89		1d5f2000	4	
40	1	1	1	0	65	4d04	09b30	00097864	7	0	89		1d5f2000	4	
41	0	0	0	0	20	0cef	09c58	0012f0c8	6	0	89		3abe4000	3	
42	0	0	0	0	20	0cef	08f69	0012f0c8	6	0	89		3abe4000	3	
43	0	0	0	0	20	0cef	0827a	0012f0c8	6	0	89		3abe4000	3	
44	0	0	0	0	21	09a1	0eb16	0025e190	5	0	89		757c8000	2	
45	0	0	0	0	21	09a1	0e175	0025e190	5	0	89		757c8000	2	
46	0	0	0	0	21	09a1	0d7d4	0025e190	5	0	89		757c8000	2	
47	0	0	0	0	21	09a1	0ce33	0025e190	5	0	89		757c8000	2	
48	0	0	0	0	21	09a1	0c492	0025e190	5	0	89		757c8000	2	
49	0	0	0	0	21	09a1	0baf1	0025e190	5	0	89		757c8000	2	
50	0	0	0	0	21	09a1	0b150	0025e190	5	0	89		757c8000	2	
51	0	0	0	0	21	09a1	0a7af	0025e190	5	0	89		757c8000	2	
52	0	0	0	0	21	09a1	09e0e	0025e190	5	0	89		757c8000	2	
53	0	0	0	0	21	09a1	0946d	0025e190	5	0	89		757c8000	2	
54	0	0	0	0	21	09a1	08acc	0025e190	5	0	89		757c8000	2	
55	0	0	0	0	21	09a1	0812b	0025e190	5	0	89		757c8000	2	
56	1	0	0	0	22	072f	0ef14	004bc320	4	0	89	89	eaf90000	1	ea

Cuadro 26 (continuación)

EC	PIX	CX	MPS	CE	ST	LSZ hex	A hex	CODIFICADOR					DECODIFICADOR		
								C hex	CT	CS	BUF hex	OUT hex	C hex	CT	IN hex
57	1	1	1	0	66	412c	0e5e0	000560a0	7	0	99		628ea000	4	
58	1	1	1	0	66	412c	0a4b4	000560a0	7	0	99		628ea000	4	
59	0	1	1	0	67	37d8	0c710	000ac140	6	0	99		c51d4000	3	
60	0	1	1	0	82	438e	0df60	002d41e0	4	0	99		d7950000	1	fa
61	0	0	0	0	46	0fbf	0871c	005bbb64	3	0	99		7786fa00	8	
62	1	0	0	0	47	09f8	0f642	00b776c8	2	0	99	99	ef0df400	7	
63	0	0	0	0	77	0d51	09f80	00063120	6	0	5c		2c3f4000	3	
64	0	0	0	0	77	0d51	0922f	00063120	6	0	5c		2c3f4000	3	
65	0	1	1	1	89	4ca9	084de	00063120	6	0	5c		2c3f4000	3	
66	0	1	1	0	95	56a8	0e0d4	0018c480	4	0	5c		b0fd0000	1	a0
67	0	1	0	0	95	56a8	0ad50	00329d58	3	0	5c		4da2a000	8	
68	1	1	0	0	96	4f46	0ad50	00653ab0	2	0	5c		9b454000	7	
69	0	1	0	1	101	5231	09e8c	00cb3174	1	0	5c	5c	7a768000	6	
70	1	1	0	0	102	4c0f	0a462	0006fb9e	8	0	32		5c370000	5	
71	1	1	0	1	106	50e7	0981e	000ea7e2	7	0	32		07c80000	4	
72	1	1	0	1	108	5597	08e6e	001d4fc4	6	0	32		0f900000	3	
73	0	0	0	0	77	0d51	0e35c	00753f10	4	0	32		3e400000	1	
74	0	0	0	0	77	0d51	0d60b	00753f10	4	0	32		3e400000	1	
75	0	0	0	0	77	0d51	0c8ba	00753f10	4	0	32		3e400000	1	
76	0	0	0	0	77	0d51	0bb69	00753f10	4	0	32		3e400000	1	
77	0	0	0	0	77	0d51	0ae18	00753f10	4	0	32		3e400000	1	
78	0	0	0	0	77	0d51	0a0c7	00753f10	4	0	32		3e400000	1	
79	0	0	0	0	77	0d51	09376	00753f10	4	0	32		3e400000	1	
80	0	0	0	0	77	0d51	08625	00753f10	4	0	32		3e400000	1	d5
81	0	0	0	0	78	0bb6	0f1a8	00ea7e20	3	0	32		7c80d500	8	
82	0	0	0	0	78	0bb6	0e5f2	00ea7e20	3	0	32		7c80d500	8	
83	0	0	0	0	78	0bb6	0da3c	00ea7e20	3	0	32		7c80d500	8	
84	0	0	0	0	78	0bb6	0ce86	00ea7e20	3	0	32		7c80d500	8	
85	0	0	0	0	78	0bb6	0c2d0	00ea7e20	3	0	32		7c80d500	8	
86	0	0	0	0	78	0bb6	0b71a	00ea7e20	3	0	32		7c80d500	8	
87	0	0	0	0	78	0bb6	0ab64	00ea7e20	3	0	32		7c80d500	8	
88	0	0	0	0	78	0bb6	09fae	00ea7e20	3	0	32		7c80d500	8	
89	0	0	0	0	78	0bb6	093f8	00ea7e20	3	0	32		7c80d500	8	
90	0	0	0	0	78	0bb6	08842	00ea7e20	3	0	32		7c80d500	8	
91	1	0	0	0	79	0a40	0f918	01d4fc40	2	0	32	32	f901aa00	7	
92	1	0	0	0	77	0d51	0a400	001eb180	6	0	ea		a29aa000	3	ff
93	0	0	0	0	75	119c	0d510	01f482f0	2	0	ea		bebbfe00	7	
94	1	0	0	0	75	119c	0c374	01f482f0	2	0	ea	ea	bebbfe00	7	
95	0	0	0	0	74	1424	08ce0	0009a640	7	0	fa		671ff000	4	
96	0	0	0	0	75	119c	0f178	00134c80	6	0	fa		ce3fe000	3	
97	0	0	0	0	75	119c	0dfdc	00134c80	6	0	fa		ce3fe000	3	
98	1	0	0	0	75	119c	0ce40	00134c80	6	0	fa		ce3fe000	3	52
99	1	0	0	0	74	1424	08ce0	00a04920	3	0	fa	fa	8cdf5200	8	
100	1	0	0	0	72	1aa9	0a120	00060ee0	8	0	a0		a11a9000	5	
101	1	0	0	0	87	2516	0d548	0034aab8	5	0	a0		d51c8000	2	7f
102	1	0	0	0	86	299a	09458	00d56ba8	3	0	a0		93aa7f00	8	
103	1	0	0	0	93	32b4	0a668	03575998	1	0	a0	a0	a3b1fc00	6	
104	1	0	0	0	99	3c3d	0cad0	000f3530	7	0	d5		bff7f000	4	
105	1	0	0	0	104	415e	0f0f4	003f0f0c	5	0	d5		c593c000	2	
106	1	0	0	1	103	4639	082bc	007f7d44	4	0	d5		2bfb8000	1	ff
107	1	0	0	0	107	4b85	0f20c	01fd510	2	0	d5		afeffe00	7	
108	1	0	0	1	109	504f	0970a	03fd372e	1	0	d5		12d1fc00	6	
109	1	0	0	1	111	5522	08d76	00026e5c	8	1	d5		25a3f800	5	
110	1	0	0	0	112	59eb	0e150	0009b970	6	1	d5		968fe000	3	
111	1	0	1	0	112	59eb	0b3d6	001481aa	5	1	d5		1e55c000	2	
112	1	0	1	0	111	5522	0b3d6	00290354	4	1	d5		3cab8000	1	ff

Cuadro 26 (continuación)

								CODIFICADOR					DECODIFICADOR		
EC	PIX	CX	MPS	CE	ST	LSZ hex	A hex	C hex	CT	CS	BUF hex	OUT hex	C hex	CT	IN hex
113	0	0	1	0	109	504f	0bd68	005206a8	3	1	d5		7957ff00	8	
114	0	0	1	1	111	5522	0a09e	00a4e782	2	1	d5		187dfe00	7	
115	0	0	1	1	112	59eb	096f8	0149cf04	1	1	d5	d5ff	30fbfc00	6	
116	1	0	0	0	112	59eb	0f434	00073c10	7	0	52		c3eff000	4	
117	1	0	1	0	112	59eb	0b3d6	000facb2	6	0	52		534de000	3	
118	0	0	1	0	111	5522	0b3d6	001f5964	5	0	52		a69bc000	2	
119	1	0	1	1	112	59eb	0aa44	003f7030	4	0	52		8fcf8000	1	ff
120	0	0	1	0	111	5522	0b3d6	007f8112	3	0	52		7eedff00	8	
121	0	0	1	1	112	59eb	0aa44	00ffb8c	2	0	52		4073fe00	7	
122	0	0	0	1	112	59eb	0a0b2	01ff7f18	1	0	52	52	80e7fc00	6	
123	1	0	0	0	111	5522	0b3d6	00078bbe	8	0	7f		7441f800	5	
124	1	0	0	1	112	59eb	0aa44	000fd4e4	7	0	7f		2b1bf000	4	
125	1	0	1	1	112	59eb	0a0b2	001fa9c8	6	0	7f		5637e000	3	
126	1	0	1	0	111	5522	0b3d6	003fe11e	5	0	7f		1ee1c000	2	
127	1	0	1	0	109	504f	0bd68	007fc23c	4	0	7f		3dc38000	1	c0
128	1	0	1	0	107	4b85	0da32	00ff8478	3	0	7f		7b87c000	8	
129	1	0	1	1	107	4b85	08ead	00ff8478	3	0	7f		7b87c000	8	
130	0	0	1	0	103	4639	0970a	01ff8f40	2	0	7f		70bf8000	7	
131	0	0	1	1	107	4b85	08c72	03ffc022	1	0	7f		3fdd0000	6	
132	1	0	1	1	109	504f	081da	00078044	8	1	7f		7fba0000	5	
133	0	0	1	0	107	4b85	0a09e	000f639e	7	1	7f		9c5e0000	4	
134	1	0	1	1	109	504f	0970a	001f716e	6	1	7f		8e8a0000	3	
135	0	0	1	0	107	4b85	0a09e	003f7052	5	1	7f		8f9e0000	2	
136	1	0	1	1	109	504f	0970a	007f8ad6	4	1	7f		750a0000	1	00
137	0	0	1	0	107	4b85	0a09e	00ffa322	3	1	7f		5c9e0000	8	
138	0	0	1	1	109	504f	0970a	01fff076	2	1	7f		0f0a0000	7	
139	0	0	1	1	111	5522	08d76	03ffe0ec	1	1	7f		1e140000	6	
140	1	0	1	0	112	59eb	0e150	000f83b0	7	2	7f		78500000	4	
141	1	0	1	1	112	59eb	08765	000f83b0	7	2	7f		78500000	4	
142	0	0	1	0	111	5522	0b3d6	001f6254	6	2	7f		95ac0000	3	
143	1	0	1	1	112	59eb	0aa44	003f8210	5	2	7f		6df00000	2	
144	1	0	1	0	111	5522	0b3d6	007fa4d2	4	2	7f		3b2e0000	1	00
145	0	0	1	0	109	504f	0bd68	00ff49a4	3	2	7f		765c0000	8	
146	0	0	1	1	111	5522	0a09e	01ff6d7a	2	2	7f		12860000	7	
147	0	0	1	1	112	59eb	096f8	03fedaf4	1	2	7f		250c0000	6	
148	0	0	0	0	112	59eb	0f434	000b6bd0	7	3	7f		94300000	4	
149	0	0	0	1	112	59eb	09a49	000b6bd0	7	3	7f		94300000	4	
150	1	0	0	0	111	5522	0b3d6	0017585c	6	3	7f		a7a40000	3	
151	0	0	0	1	112	59eb	0aa44	002f6e20	5	3	7f		91e00000	2	
152	1	0	0	0	111	5522	0b3d6	005f7cf2	4	3	7f		830e0000	1	00
153	1	0	0	1	112	59eb	0aa44	00bf74c	3	3	7f		48b40000	8	
154	1	0	1	1	112	59eb	0a0b2	017f6e98	2	3	7f		91680000	7	
155	0	0	1	0	111	5522	0b3d6	02ff6abe	1	3	7f	7fffffff	95420000	6	
156	1	0	1	1	112	59eb	0aa44	000792e4	8	0	bf		6d1c0000	5	
157	1	0	1	0	111	5522	0b3d6	000fc67a	7	0	bf		39860000	4	
158	0	0	1	0	109	504f	0bd68	001f8cf4	6	0	bf		730c0000	3	
159	0	0	1	1	111	5522	0a09e	003ff41a	5	0	bf		0be60000	2	
160	0	0	1	1	112	59eb	096f8	007fe834	4	0	bf		17cc0000	1	3f
161	0	0	0	0	112	59eb	0f434	01ffa0d0	2	0	bf		5f307e00	7	
162	0	0	0	1	112	59eb	09a49	01ffa0d0	2	0	bf		5f307e00	7	
163	0	0	0	0	111	5522	0b3d6	03ffc25c	1	0	bf		3da4fc00	6	
164	1	0	0	0	109	504f	0bd68	000784b8	8	1	bf		7b49f800	5	
165	1	0	0	1	111	5522	0a09e	000fe3a2	7	1	bf		1c61f000	4	
166	1	0	0	1	112	59eb	096f8	001fc744	6	1	bf		38c3e000	3	
167	0	0	1	0	112	59eb	0f434	007f1d10	4	1	bf		e30f8000	1	ff
168	1	0	0	0	112	59eb	0b3d6	00ff6eb2	3	1	bf		918dff00	8	

Cuadro 26 (continuación)

EC	PIX	CX	MPS	CE	ST	LSZ hex	A hex	CODIFICADOR					DECODIFICADOR		
								C hex	CT	CS	BUF hex	OUT hex	C hex	CT	IN hex
169	0	0	1	0	112	59eb	0b3d6	01ff913a	2	1	bf		6f45fe00	7	
170	0	0	0	0	112	59eb	0b3d6	03ffd64a	1	1	bf		2ab5fc00	6	
171	0	0	0	0	111	5522	0b3d6	0007ac94	8	2	bf		556bf800	5	
172	1	0	0	0	109	504f	0bd68	000f5928	7	2	bf		aad7f000	4	
173	0	0	0	1	111	5522	0a09e	001f8c82	6	2	bf		7b7de000	3	
174	1	0	0	0	109	504f	0aa44	003faffc	5	2	bf		6003c000	2	
175	1	0	0	1	111	5522	0a09e	008013e2	4	2	bf		0c1d8000	1	2d
176	1	0	0	1	112	59eb	096f8	010027c4	3	2	bf		183b2d00	8	
177	1	0	1	0	112	59eb	0f434	04009f10	1	2	bf	c00000	60ecb400	6	
178	1	0	1	1	112	59eb	09a49	04009f10	1	2	bf		60ecb400	6	
179	1	0	1	0	111	5522	0b3d6	0001bedc	8	0	00		411d6800	5	
180	0	0	1	0	109	504f	0bd68	00037db8	7	0	00		823ad000	4	
181	0	0	1	1	111	5522	0a09e	0007d5a2	6	0	00		2a43a000	3	
182	1	0	1	1	112	59eb	096f8	000fab44	5	0	00		54874000	2	
183	1	0	1	0	111	5522	0b3d6	001fd0a2	4	0	00		2ef48000	1	20
184	1	0	1	0	109	504f	0bd68	003fa144	3	0	00		5de92000	8	
185	0	0	1	0	107	4b85	0da32	007f4288	2	0	00		bbd24000	7	
186	1	0	1	1	109	504f	0970a	00ffa26a	1	0	00	00	5a4a8000	6	
187	1	0	1	0	107	4b85	0a09e	0007d24a	8	0	3f		271f0000	5	
188	1	0	1	0	103	4639	0aa32	000fa494	7	0	3f		4e3e0000	4	
189	0	0	1	0	104	415e	0c7f2	001f4928	6	0	3f		9c7c0000	3	
190	0	0	1	1	103	4639	082bc	003f9f78	5	0	3f		2b000000	2	82
191	0	0	1	0	107	4b85	0f20c	00fe7de0	3	0	3f		af408200	8	
192	0	0	1	1	109	504f	0970a	01fe48ce	2	0	3f		11730400	7	
193	0	0	1	1	111	5522	08d76	03fc919c	1	0	3f		22e60800	6	
194	0	0	1	0	112	59eb	0e150	00024670	7	1	3f		8b982000	4	
195	0	0	0	0	112	59eb	0b3d6	00059baa	6	1	3f		08664000	3	
196	0	0	0	0	111	5522	0b3d6	000b3754	5	1	3f		10cc8000	2	
197	0	0	0	0	109	504f	0bd68	00166ea8	4	1	3f		21990000	1	91
198	0	0	0	0	107	4b85	0da32	002cdd50	3	1	3f		43329100	8	
199	0	0	0	1	107	4b85	08ead	002cdd50	3	1	3f		43329100	8	
200	0	0	0	0	103	4639	0970a	005a40f0	2	1	3f		00152200	7	
201	0	0	0	0	104	415e	0a1a2	00b481e0	1	1	3f	3fff	002a4400	6	
202	0	0	0	0	99	3c3d	0c088	000103c0	8	0	2d		00548800	5	
203	0	0	0	0	99	3c3d	0844b	000103c0	8	0	2d		00548800	5	
204	0	0	0	0	100	375e	0901c	00020780	7	0	2d		00a91000	4	
205	0	0	0	0	93	32b4	0b17c	00040f00	6	0	2d		01522000	3	
206	0	0	0	0	94	2e17	0fd90	00081e00	5	0	2d		02a44000	2	
207	0	0	0	0	94	2e17	0cf79	00081e00	5	0	2d		02a44000	2	
208	0	0	0	0	94	2e17	0a162	00081e00	5	0	2d		02a44000	2	
209	0	0	0	0	86	299a	0e696	00103c00	4	0	2d		05488000	1	
210	0	0	0	0	86	299a	0bcfc	00103c00	4	0	2d		05488000	1	
211	0	0	0	0	86	299a	09362	00103c00	4	0	2d		05488000	1	00
212	0	0	0	0	87	2516	0d390	00207800	3	0	2d		0a910000	8	
213	0	0	0	0	87	2516	0ae7a	00207800	3	0	2d		0a910000	8	
214	0	0	0	0	87	2516	08964	00207800	3	0	2d		0a910000	8	
215	0	0	0	0	71	1edf	0c89c	0040f000	2	0	2d		15220000	7	
216	0	0	0	0	71	1edf	0a9bd	0040f000	2	0	2d		15220000	7	
217	0	0	0	0	71	1edf	08ade	0040f000	2	0	2d		15220000	7	
218	0	0	0	0	72	1aa9	0d7fe	0081e000	1	0	2d		2a440000	6	
219	0	0	0	0	72	1aa9	0bd55	0081e000	1	0	2d		2a440000	6	
220	0	0	0	0	72	1aa9	0a2ac	0081e000	1	0	2d		2a440000	6	
221	0	0	0	0	72	1aa9	08803	0081e000	1	0	2d	2d	2a440000	6	
222	0	0	0	0	73	174e	0dab4	0003c000	8	0	20		54880000	5	
223	0	0	0	0	73	174e	0c366	0003c000	8	0	20		54880000	5	

Cuadro 26 (fin)

EC	PIX	CX	MPS	CE	ST	LSZ hex	A hex	CODIFICADOR					DECODIFICADOR		
								C hex	CT	CS	BUF hex	OUT hex	C hex	CT	IN hex
224	0	0	0	0	73	174e	0ac18	0003c000	8	0	20		54880000	5	
225	0	0	0	0	73	174e	094ca	0003c000	8	0	20		54880000	5	
226	0	0	0	0	74	1424	0faf8	00078000	7	0	20		a9100000	4	
227	0	0	0	0	74	1424	0e6d4	00078000	7	0	20		a9100000	4	
228	0	0	0	0	74	1424	0d2b0	00078000	7	0	20		a9100000	4	
229	0	0	0	0	74	1424	0be8c	00078000	7	0	20		a9100000	4	
230	1	0	0	0	74	1424	0aa68	00078000	7	0	20		a9100000	4	
231	1	0	0	0	72	1aa9	0a120	0040b220	4	0	20		96600000	1	00
232	0	0	0	0	87	2516	0d548	0209c4b8	1	0	20		7f480000	6	
233	0	0	0	0	87	2516	0b032	0209c4b8	1	0	20		7f480000	6	
234	1	0	0	0	87	2516	08b1c	0209c4b8	1	0	20	20	7f480000	6	
235	0	0	0	0	86	299a	09458	0008aaf8	7	0	82		65080000	4	
236	1	0	0	0	87	2516	0d57c	001155f0	6	0	82		ca100000	3	
237	0	0	0	0	86	299a	09458	00481958	4	0	82		66a80000	1	00
238	1	0	0	0	87	2516	0d57c	009032b0	3	0	82		cd500000	8	
239	1	0	0	0	86	299a	09458	02438c58	1	0	82	82	73a80000	6	
240	0	0	0	0	93	32b4	0a668	000fdc58	7	0	90		23a80000	4	
241	0	0	0	0	94	2e17	0e768	001fb8b0	6	0	90		47500000	3	
242	0	0	0	0	94	2e17	0b951	001fb8b0	6	0	90		47500000	3	
243	0	0	0	0	94	2e17	08b3a	001fb8b0	6	0	90		47500000	3	
244	0	0	0	0	86	299a	0ba46	003f7160	5	0	90		8ea00000	2	
245	1	0	0	0	86	299a	090ac	003f7160	5	0	90		8ea00000	2	00
246	1	0	0	0	93	32b4	0a668	00ff61c8	3	0	90		9e380000	8	
247	1	0	0	0	99	3c3d	0cad0	03ff55f0	1	0	90		aa100000	6	
248	0	0	0	0	104	415e	0f0f4	000f920c	7	1	90		6df40000	4	
249	0	0	0	0	104	415e	0af96	000f920c	7	1	90		6df40000	4	
250	1	0	0	0	99	3c3d	0dc70	001f2418	6	1	90		dbe80000	3	
251	1	0	0	0	104	415e	0f0f4	007f112c	4	1	90		eed40000	1	00
252	0	0	0	1	103	4639	082bc	00ff8184	3	1	90		7e7c0000	8	
253	1	0	0	0	104	415e	08c72	01ff7c0e	2	1	90		83f20000	7	
254	0	0	0	1	103	4639	082bc	03ff8e44	1	1	90		71bc0000	6	
255	1	0	0	0	104	415e	08c72	0007958e	8	2	90		6a720000	5	
256	0	0	0	1	103	4639	082bc	000fc144	7	2	90		3ebc0000	4	
257							08c72	08000000	6	2	90	91			

7.2 Algoritmo parametrizado

Esta subcláusula normativa describe métodos de prueba del algoritmo descrito en cláusulas anteriores de esta Especificación. Existen muchas parametrizaciones posibles, y esta subcláusula describe maneras de probar la exactitud de algunas de ellas, consideradas de utilidad para depurar las realizaciones. Si se pretende que la realización de un codificador admite una parametrización igual o más amplia que la de cualquiera de las configuraciones para las que se suministran datos de prueba en esta subcláusula, dicha realización deberá generar exactamente el cómputo de bytes indicado para los datos de prueba. Si se pretende que la realización de un decodificador admite una parametrización igual o más amplia que la de cualquier configuración para la que se suministran datos de prueba en esta subcláusula dicha realización deberá decodificar esos datos de prueba (generados por una realización de codificador que cumple los requisitos de realización de codificador descritos anteriormente) y generar exactamente la imagen artificial descrita en 7.2.1. Un codificador que admita AT pero que no utilice el algoritmo propuesto en el Anexo C para determinar el desplazamiento de los pixels AT forzará artificialmente movimientos AT idénticos a los que aquí se describen. Asimismo, un codificador que no elija retirar todos los bytes 0x00 posibles del final de todas las SDE tendrá que llevar a cabo temporalmente un posprocesamiento para ello a fin de duplicar el cómputo de bytes dado.

7.2.1 Imagen artificial

Las distintas pruebas del algoritmo completo utilizan una imagen generada artificialmente. Esta imagen es generada conforme al diagrama de flujo de la Figura 38. Tiene 1960 pixels/línea, consta de 1951 líneas y contiene 861965 pixels de primer plano y 2961995 pixels de fondo.

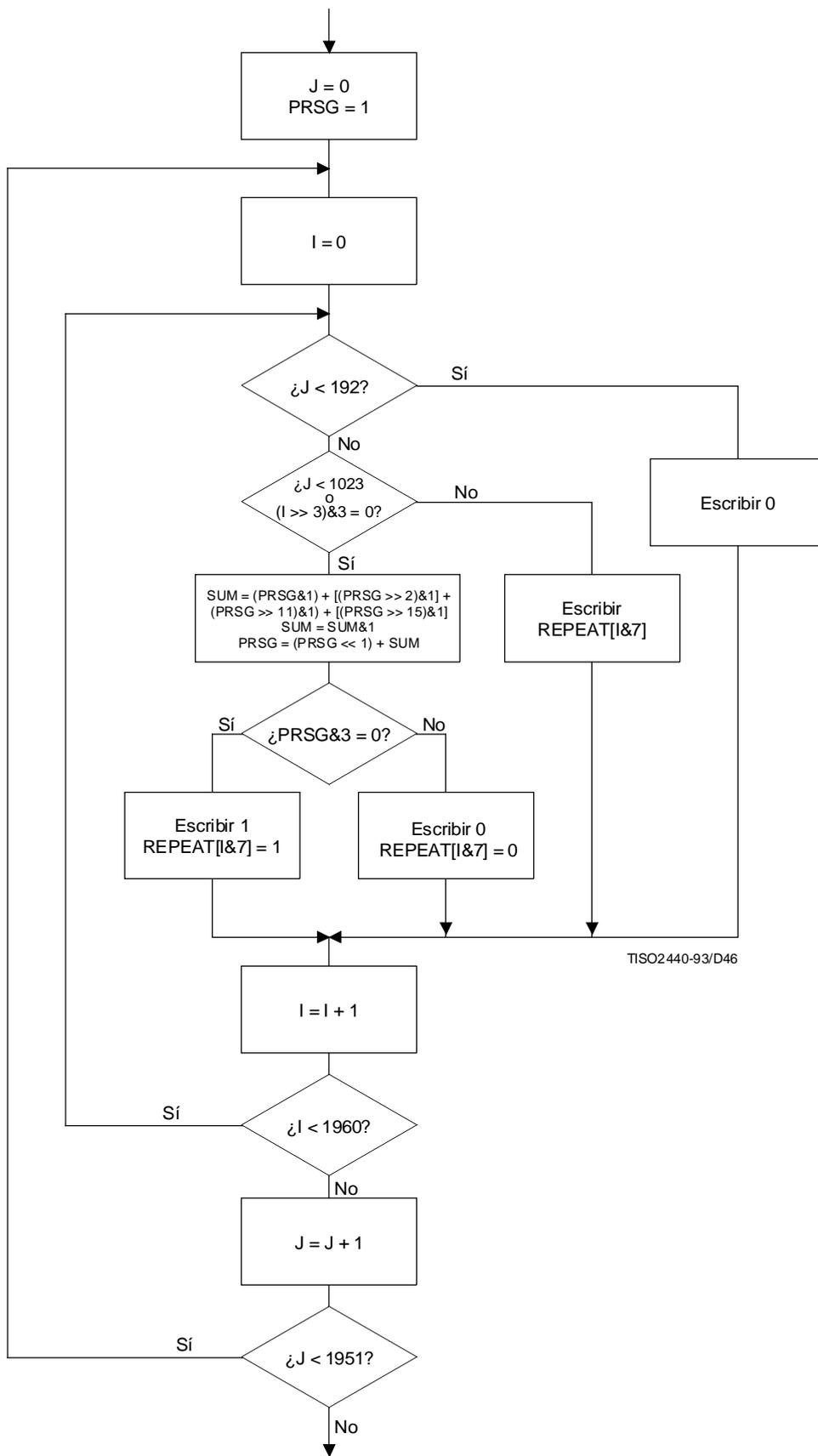


Figura 38 – Procedimiento para generar la imagen de prueba

Esta imagen ha sido construida así con el fin de emplear la mayor cantidad de características posible. De ninguna manera es una imagen típica, y los resultados de su compresión no son representativos.

7.2.2 Pruebas secuenciales de una sola progresión

Para las tres pruebas de este punto, $D_L = 0$, $D = 0$, $P = 1$, $X_0 = 1960$, $Y_0 = 1951$ y $M_Y = 0$. Los valores de HITOLO, SEQ, ILEAVE, SMID, VLENGTH, TPDON, DPON, DPPRIV, DPLAST son irrelevantes. Los cuatro parámetros restantes, L_0 , M_X , LRLTWO y TPBON varían como se muestra en el Cuadro 27. Las demás columnas de este cuadro proporcionan datos de seguimiento cuando la imagen de entrada es la imagen artificial descrita en el punto anterior. Para cada una de las dos primeras pruebas hay sólo un SCD, y el cómputo de bytes indicado es su tamaño. La prueba final, con $L_0 = 128$, produce 16 SCD, y el cómputo de bytes indicado es la suma de sus tamaños. En todos los casos, todos los bytes 0x00 restantes posibles son retirados del final de cada SCD (véanse 6.8.2.10 y la Figura 28).

Cuadro 27 – Parámetros de seguimiento para pruebas de codificación secuencial de una sola progresión

TPBON	M_X	LRLTWO	L_0	Pixels TP	Pixels codificados	Bytes codificados
0	0	0	1951	0	3 823 960	316 094
0	0	1	1951	0	3 823 960	315 887
1	8	0	128	376 320	3 447 640	252 557

En las dos primeras pruebas, la AT es efectivamente neutralizada al poner M_X igual a cero. La última prueba activa la AT así como la TP de capa de resolución mínima. AT se realizó como se describe en el Anexo C, y se atendió la sugerencia de diferir cualquier conmutación de AT hasta el comienzo de la barra siguiente. Los datos en la última línea del Cuadro 28 se obtuvieron utilizando SDNORM, lo que significa que el estimador de probabilidad no se ha reinicializado. En el Cuadro 29 se proporciona información útil para depurar AT. Las dos primeras columnas dan la barra y la línea en que se hace efectiva la conmutación (tanto la numeración de las líneas como la de las barras comienza por cero), y la tercera columna da el intervalo τ_X para la nueva ubicación de pixel AT. Las 8 columnas finales dan los valores de los contadores descritos en el Anexo C cuando se dispara el desplazamiento de AT.

Cuadro 28 – Información de cambio de AT para la tercera prueba de codificación secuencial de una sola progresión

Barra	Línea	τ_X	C_{all}	C_0	C_3	C_4	C_5	C_6	C_7	C_8
9	0	8	3900	2336	2456	2472	2446	2442	2730	3534

Cuadro 29 – Cómputo de bytes para pruebas de codificación secuencial de una sola progresión

TPBON	M_X	LRLTWO	L_0	SCD	PSCD	SDE	BID	BIE
0	0	0	1951	316 094	317 362	317 364	317 364	317 384
0	0	1	1951	315 887	317 110	317 112	317 112	317 132
1	8	0	128	252 557	253 593	253 625	253 633	253 653

En el Cuadro 30 se proporcionan más datos para estas tres pruebas. Las entradas de la columna SCD duplican la última columna del Cuadro 27. Los datos codificados de barra protegidos (PSCD, *protected stripe coded data*) se obtienen reemplazando cada byte alineado 0xff por 0xff (ESC) y 0x00 (STUFF) en los datos codificados de barra (SCD, *stripe coded data*). La entidad de datos de barra (SDE, *stripe data entity*) se obtiene añadiendo un byte 0xff (ESC) y un byte 0x02 (SDNORM) al final de cada PSCD. Los datos de la imagen binaria tienen el mismo número de bytes que la SDE, excepto para la última prueba, en la que tienen ocho bytes más, debido al segmento marcador de desplazamiento AT. Finalmente, la BIE es 20 bytes más grande (el tamaño del encabezamiento) que BID.

Cuadro 30 – Parámetros de seguimiento para la codificación de la imagen artificial

Capa	X_d	Y_d	L_d	Líneas TP	Excepciones TP	Pixels TP	Pixels DP	Pixels codificados	Bytes codificados
6	1960	1951	128	137	7033	375 520	589 344	2 859 096	188 817
5	980	976	64	186	1442	93 120	128 642	734 718	65 584
4	490	488	32	181	135	22 792	30 230	186 098	16 565
3	245	244	16	117	8	5406	7246	47 128	4994
2	123	122	8	61	0	1238	1769	11 999	1430
1	62	61	4	31	0	248	452	3082	370
0	31	31	2	3	–	93	–	868	113

7.2.3 Prueba progresiva y secuencial compatible progresiva

Para la prueba de este punto, la imagen de entrada es una vez más la imagen artificial, pero esta vez los parámetros tienen los valores siguientes (véase el Cuadro 9): $D_L = 0$, $D = 6$, $P = 1$, $X_6 = 1960$, $Y_6 = 1951$, $L_0 = 2$, $M_X = 8$, $M_Y = 0$, HITOLO = 0 (irrelevante), SEQ = 0 (irrelevante), ILEAVE = 0 (irrelevante), SMID = 0 (irrelevante), LRLTWO = 0, VLENGTH = 0 (irrelevante), TPDON = 1, TPBON = 1, DPPRIV = 0 y DPLAST = 0 (irrelevante). Si bien la parametrización anterior es una parametrización de codificación progresiva, el valor de SEQ es irrelevante y se generan cómputos de bytes idénticos en codificación secuencial compatible progresiva. Por consiguiente, los datos de prueba de este caso pertenecen a ambos modos. AT es una vez más realizada como se sugiere en el Anexo C, siendo conveniente diferir cualquier conmutación hasta el comienzo de la barra siguiente.

El Cuadro 31 proporciona datos de seguimiento para codificar la imagen artificial con los parámetros anteriores. El cómputo de bytes codificados que se indica en la última columna es la suma del número de bytes de los 16 SCD de esa capa. Como antes, se retiran todos los bytes 0x00 posibles.

Hay dos conmutaciones de plantilla adaptativa, una en la capa 6 y una en la capa 5. En el Cuadro 31 se proporcionan los datos correspondientes a estas dos conmutaciones.

Cuadro 31 – Información de cambio de AT

Capa	Barra	Línea	τ_X	C_{all}	C_0	C_3	C_4	C_5	C_6	C_7	C_8
6	9	0	8	3243	1984	2014	2055	2031	2001	2212	2924
5	10	0	4	2580	1323	1401	2259	1440	1447	1426	1966

En el Cuadro 32 figura el cómputo de bytes por capa para los 16 PSCD y SDE, así como el número total de bytes en BID y BIE (véase 6.2). PSCD se obtiene reemplazando cada byte alineado 0xff por 0xff (ESC) y 0x00 (STUFF) en SCD. La entidad de datos de barra (SDE, *stripe data entity*) se obtiene añadiendo un byte 0xff (ESC) y un byte 0x02 (SDNORM) al final de cada PSCD. Finalmente, el campo de datos de imagen binaria BID se obtiene concatenando las 112 SDE (7 capas con 16 barras cada una) y los dos segmentos marcadores ATMOVE, y la BIE se obtiene añadiendo los BID al BIH de veinte bytes.

Cuadro 32 – Cómputo de bytes para la imagen artificial

Capa	SCD	PSCD	SDE	BID	BIE
6	188 817	189 584	189 616		
5	65 584	65 905	65 937		
4	16 565	16 634	16 666		
3	4994	5010	5042		
2	1430	1434	1466		
1	370	373	405		
0	113	114	146		
Total	277 873	279 054	279 278	279 294	279 314

7.3 Ejemplos de tren de datos

Ejemplo de BIE para la codificación secuencial de una sola progresión de una imagen binaria de 1728 pixels de anchura y 2376 de altura con una barra para la imagen entera y todos los parámetros binarios puestos a cero (en hexadecimal):

```
|0x00|0x01|0x01|0x00|0x00 0x00 0x06 0xc0|0x00 0x00 0x09 0x48|
|0x00 0x00 0x09 0x48|0x00|0x00|0x00|0x00|
| PSCD para la imagen entera |0xff|0x02|
```

(Las barras verticales indican agrupamientos lógicos, pero no son distintas de un simple espacio.)

Ejemplo de BIE para la codificación secuencial compatible progresiva de una imagen binaria de 1728 pixels de anchura y 2376 pixels de altura con una capa diferencial, 64 líneas por barra en la imagen reducida y todos los parámetros binarios, excepto SEQ, puestos a cero (en hexadecimal):

```
|0x00|0x01|0x01|0x00|0x00 0x00 0x06 0xc0|0x00 0x00 0x09 0x48|
|0x00 0x00 0x00 0x40|0x00|0x00|0x04|0x00|
| PSCD para la barra 0 y la capa 0 (líneas de imagen de base 0 a 63)|0xff|0x02|
| PSCD para la barra 0 y la capa 1 (líneas de imagen diferencial 0 a 127)|0xff|0x02|
| PSCD para la barra 1 y la capa 0 (líneas de imagen de base 64 a 127)|0xff|0x02|
| PSCD para la barra 1 y la capa 1 (líneas de imagen diferencial 128 a 255)|0xff|0x02|
.
.
.
| PSCD para la barra 17 y la capa 0 (líneas de imagen de base 1088 a 1151)|0xff|0x02|
| PSCD para la barra 17 y la capa 1 (líneas de imagen diferencial 2176 a 2303)|0xff|0x02|
| PSCD para la barra 18 y la capa 0 (líneas de imagen de base 1152 a 1187)|0xff|0x02|
| PSCD para la barra 18 y la capa 1 (líneas de imagen diferencial 2304 a 2375)|0xff|0x02|
```

Los comentarios que figuran entre paréntesis indican las líneas de las imágenes de base y diferencial que se codifican en cada barra.

Ejemplo de BIE para la codificación progresiva de una imagen binaria de 1728 pixels de anchura y 2376 pixels de altura con una capa diferencial, 64 líneas por barra en la imagen reducida y todos los parámetros binarios puestos a cero (en hexadecimal):

```
|0x00|0x01|0x01|0x00|0x00 0x00 0x06 0xc0|0x00 0x00 0x09 0x48|
|0x00 0x00 0x00 0x40|0x00|0x00|0x00|0x00|
| PSCD para la barra 0 y la capa 0 (líneas de imagen de base 0 a 63)|0xff|0x02|
| PSCD para la barra 1 y la capa 0 (líneas de imagen de base 64 a 127)|0xff|0x02|
.
.
.
| PSCD para la barra 17 y la capa 0 (líneas de imagen de base 1088 a 1151)|0xff|0x02|
| PSCD para la barra 18 y la capa 0 (líneas de imagen de base 1152 a 1187)|0xff|0x02|
| PSCD para la barra 0 y la capa 1 (líneas de imagen diferencial 0 a 127)|0xff|0x02|
| PSCD para la barra 1 y la capa 1 (líneas de imagen diferencial 128 a 255)|0xff|0x02|
.
.
.
| PSCD para la barra 17 y la capa 1 (líneas de imagen diferencial 2176 a 2303)|0xff|0x02|
| PSCD para la barra 18 y la capa 1 (líneas de imagen diferencial 2304 a 2375)|0xff|0x02|
```

Anexo A

Soporte mínimo sugerido de los parámetros libres

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

Las aplicaciones pueden fijar cualquiera de los 19 parámetros libres a cualesquiera valores de las gamas especificadas en el Cuadro 9. Las sugerencias sobre la sustentación mínima contenida en este anexo se dan a fin de permitir la posibilidad de una amplia gama de aplicaciones que compartan el soporte físico e intercambien datos de imagen decodificables. A menos que sea absolutamente necesario, se recomienda que las aplicaciones no elijan valores de parámetros fuera de las gamas de soporte sugeridas.

El Cuadro A.1 indica las mínimas gamas de soporte sugeridas. Los dispositivos de soporte físico y soporte lógico destinados a utilización general deben ser tales que si se dotan de suficiente memoria externa existe soporte para opciones de parámetros dentro de las gamas indicadas. Un decodificador que incluye un dispositivo de salida específico debe proporcionar soporte en esas mismas gamas, pero con las excepciones obvias de que no necesita soporte para:

- dimensiones de imagen X_D e Y_D superiores a la capacidad del dispositivo de salida concreto disponible,
- valores de P superiores a la capacidad del dispositivo de salida concreto disponible, y
- más de un orden de barra definido por SEQ, ILEAVE y SMID.

NOTA – En las aplicaciones secuenciales de una sola progresión, la sustentación máxima de D es cero, la sustentación máxima de L_0 es Y_D , y no se requiere sustentación de los parámetros HITOLO, SEQ, TPDON, DPON, DPPRIV y DPLAST.

Cuadro A.1 – Sustentación mínima sugerida de los parámetros libres

Parámetro	Mínimo	Máximo
D_L	0	D
D	D_L	6
P	1	4
X_D	1	5184
Y_D	1	8192
L_0	1	$128/2^D$, para $D > 0$ Y_D , para $D = 0$
M_X	0	8, para codificadores 16, para decodificadores
M_Y	0	0
HITOLO	0	0
SEQ	0	1
ILEAVE	0	1
SMID	0	1
LRLTWO	0	1
VLENGTH	0	1
TPDON	0	1
TPBON	0	1
DPON	0	1
DPPRIV	0	1
DPLAST	0	1

Anexo B

Diseño de la tabla de reducción de resolución

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

B.1 Filtrado

El principio que fundamenta el algoritmo de reducción es la preservación de densidad mediante el uso de filtrado (una ecuación diferencial). Sin embargo, es necesario revocar en ocasiones la salida del filtro en aras de la preservación de bordes, preservación de líneas, preservación de patrones periódicos y preservación de patrones de tremolación. Estas revocaciones de la regla general se denominan «excepciones».

La Figura B.1 es una repetición de la Figura 4, pero con cambios en los nombres de los diversos pixels. El color del pixel objetivo «?» se decide utilizando no sólo los cuatro pixels de baja resolución correspondientes $h_{22}, h_{23}, h_{32}, h_{33}$, sino también los pixels periféricos $h_{11}, h_{12}, h_{13}, h_{21}, h_{31}$ y los pixels de baja resolución ya comprometidos l_{00}, l_{01}, l_{10} . Si el valor de l_{00} no es igual al valor de h_{11} , su diferencia de valor afecta a la decisión vigente sobre el valor de l_{11} . Análogamente, las diferencias de valor entre l_{10} y h_{21} , entre l_{10} y h_{31} , entre l_{01} y h_{12} , y entre l_{01} y h_{13} afectan también a la decisión del valor vigente. Específicamente, se forma la magnitud

$$4h_{22} + 2(h_{23} + h_{32}) + h_{33} + (h_{11} - l_{00}) + 2(h_{21} - l_{10}) + (h_{31} - l_{10}) + 2(h_{12} - l_{01}) + (h_{13} - l_{01}) \quad (B-1)$$

o, su equivalente

$$4h_{22} + 2(h_{12} + h_{21} + h_{23} + h_{32}) + (h_{11} + h_{13} + h_{31} + h_{33}) - 3(l_{01} + l_{10}) - l_{00} \quad (B-2)$$

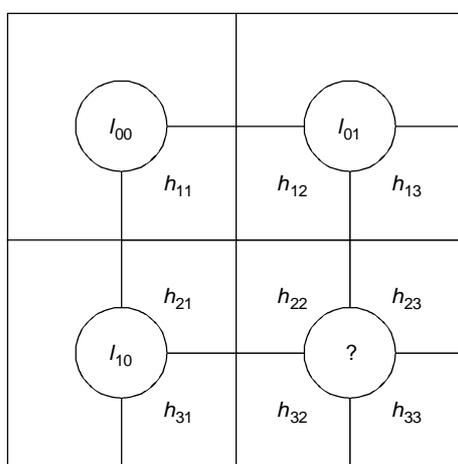


Figura B.1 – Pixels utilizados para determinar el color de un pixel de baja resolución

La Figura B.2 muestra gráficamente las ponderaciones de la última forma de la expresión.

Suponiendo que el blanco y el negro son igualmente probables, y que los pixels son estadísticamente independientes, el valor esperado de la expresión B-2 es 4,5. El pixel l_{11} se decide provisionalmente que sea 1, si y sólo si la expresión B-2 es mayor que 4,5. (Debe ser entero.)

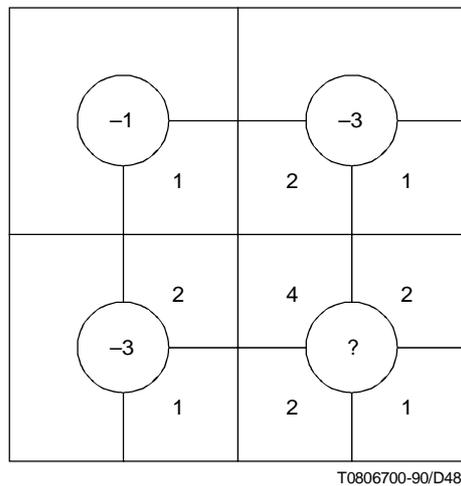


Figura B.2 – Ponderaciones de pixels

B.2 Excepciones

B.2.1 Preservación de bordes

La preservación de bordes es la más básica de las motivaciones para introducir excepciones. Las excepciones de bordes definidas ayudan a efectuar bordes de regiones de color continuas rectos y no en zigzag, sin que importe que el borde se produzca en una línea de alta resolución par o impar o en una columna de alta resolución par o impar. Los contextos de las 132 excepciones de bordes se indican a continuación en hexadecimal. Los colores de pixel se hacen corresponder con estos contextos enteros mediante la correspondencia de significación de bits de la Figura 4. Para los 132 contextos enumerados, se invierte el color real elegido con respecto al dado por la regla general.

```
0x007 0x207 0x407 0x607 0x807 0xa07 0xc07 0xe07 0x20f 0x40f 0x60f 0xa0f
0xc0f 0xe0f 0x617 0xa17 0xc17 0xe17 0x61f 0xe1f 0x227 0x427 0x627 0xa27
0xc27 0xe27 0x62f 0xa2f 0xc2f 0xe2f 0x637 0xe37 0x247 0x447 0x647 0x847
0xa47 0xc47 0xe47 0x049 0x249 0x449 0x649 0x849 0xa49 0xc49 0xe49 0x24b
0x44b 0x64b 0xa4b 0xc4b 0xe4b 0x24d 0x44d 0x64d 0x84d 0xa4d 0xc4d 0xe4d
0x659 0xa59 0xc59 0xe59 0x65b 0xe5b 0x269 0x469 0x669 0xa69 0xc69 0xe69
0x287 0x487 0x687 0xa87 0xc87 0xe87 0x8b6 0xab6 0x2c9 0x4c9 0x6c9 0xac9
0xcc9 0xec9 0x6cb 0xacb 0xccb 0xecb 0x6d9 0xed9 0x8f8 0xcf8 0x307 0x507
0x707 0x907 0xb07 0xd07 0xf07 0x334 0x336 0xb36 0x349 0x549 0x749 0x949
0xb49 0xd49 0xf49 0x578 0xd78 0x396 0xb96 0x3a6 0xbb2 0x3b4 0xbb4 0x3b6
0xbb6 0xfb6 0xdb8 0x5d0 0x5d8 0xdd8 0x5e8 0x5f0 0xdf0 0x5f8 0xdf8 0xff8
```

B.2.2 Preservación de líneas

Las líneas verticales y horizontales se preservan por la regla general y por las excepciones de bordes. Las excepciones de línea definidas ayudan a preservar el encadenamiento entre líneas oblicuas. Las excepciones de líneas son importantes para la calidad de las imágenes de texto. A continuación se indican las 420 excepciones de líneas.

```
0x003 0x009 0x00a 0x022 0x088 0x0a6 0x0e8 0x0ee 0x116 0x124 0x158 0x168
0x170 0x186 0x194 0x1a2 0x1a8 0x1c0 0x20a 0x20e 0x215 0x21c 0x21d 0x223
0x22a 0x22e 0x231 0x235 0x236 0x24a 0x24f 0x252 0x254 0x270 0x277 0x289
0x291 0x294 0x2a4 0x2a8 0x2dd 0x2e0 0x2ed 0x2ee 0x311 0x312 0x318 0x322
```

ISO/CEI 11544:1993(S)

0x323 0x327 0x350 0x35b 0x35f 0x36b 0x36e 0x376 0x377 0x388 0x394 0x3ab
0x3ad 0x3c0 0x3c1 0x3c2 0x3c4 0x3c8 0x3c9 0x3dd 0x3e0 0x3f5 0x40a 0x40e
0x41c 0x423 0x431 0x44a 0x44f 0x451 0x452 0x453 0x454 0x470 0x477 0x489
0x48a 0x491 0x494 0x4a2 0x4a4 0x4ca 0x4d1 0x4d8 0x4dd 0x4e0 0x4e7 0x4ee
0x511 0x512 0x514 0x518 0x51f 0x522 0x524 0x525 0x526 0x527 0x52c 0x55f
0x564 0x570 0x577 0x588 0x589 0x58f 0x5a4 0x5ab 0x5c9 0x5ce 0x5dc 0x5e3
0x5f5 0x60e 0x61c 0x623 0x631 0x64a 0x64f 0x652 0x657 0x65d 0x66b 0x66f
0x670 0x677 0x678 0x679 0x689 0x68f 0x691 0x694 0x696 0x697 0x6a7 0x6b0
0x6b1 0x6b2 0x6b4 0x6b8 0x6cf 0x6dd 0x6e9 0x6f0 0x711 0x712 0x717 0x718
0x722 0x725 0x726 0x727 0x72c 0x72f 0x733 0x734 0x735 0x736 0x737 0x738
0x73c 0x757 0x759 0x75d 0x75f 0x764 0x776 0x777 0x788 0x792 0x799 0x7a6
0x7a7 0x7b4 0x7c1 0x7c2 0x7c4 0x7c8 0x7c9 0x7cb 0x7d0 0x7d1 0x7d2 0x7d8
0x7d9 0x7dc 0x7dd 0x7e8 0x7e9 0x7f0 0x7f5 0x80e 0x822 0x823 0x826 0x830
0x84a 0x866 0x86d 0x888 0x889 0x890 0x8a4 0x8c7 0x8c8 0x8cc 0x8e0 0x8ee
0x90f 0x916 0x922 0x924 0x925 0x947 0x94b 0x94d 0x94f 0x958 0x964 0x968
0x969 0x970 0x986 0x987 0x988 0x994 0x9b0 0x9c0 0x9c1 0x9c4 0xa0a 0xa0e
0xa1c 0xa23 0xa2a 0xa2e 0xa31 0xa32 0xa35 0xa4a 0xa4f 0xa52 0xa54 0xa56
0xa70 0xa74 0xa77 0xa88 0xa89 0xa8f 0xa91 0xa94 0xaa4 0xaa7 0xadd 0xae0
0xae2 0xae4 0xae8 0xaed 0xaee 0xb11 0xb12 0xb13 0xb18 0xb22 0xb27 0xb2e
0xb31 0xb5b 0xb6b 0xb6e 0xb76 0xb88 0xb89 0xb91 0xba8 0xbab 0xbac 0xbad
0xbb0 0xbb5 0xbc0 0xbc1 0xbc2 0xbc4 0xbc8 0xbc9 0xbd0 0bdd 0xbe0 0be2
0xbe4 0xbe8 0xbf5 0xc0a 0xc0e 0xc1c 0xc22 0xc23 0xc31 0xc4a 0xc4f 0xc52
0xc54 0xc5c 0xc6b 0xc70 0xc77 0xc88 0xc89 0xc8a 0xc91 0xc94 0xc98 0xca4
0xca6 0xcac 0cca 0xcd1 0xcd4 0xcd 0xce0 0xce4 0xce7 0xce9 0cee 0xd11
0xd12 0xd18 0xd19 0xd1f 0xd22 0xd23 0xd24 0xd25 0xd26 0xd27 0xd2c 0xd31
0xd34 0xd64 0xd77 0xd88 0xd8f 0xd91 0xda2 0xda4 0xda6 0xdab 0xdac 0xdb0
0xdc9 0xdca 0xdce 0xddc 0xde2 0xde3 0xde4 0xdf1 0xdf5 0xe0e 0xe1c 0xe23
0xe31 0xe4a 0xe4f 0xe52 0xe57 0xe5d 0xe5f 0xe6b 0xe6f 0xe70 0xe79 0xe89
0xe8f 0xe91 0xe94 0xe97 0xea4 0xea7 0xecf 0xee0 0xee9 0xeef 0xf11 0xf12
0xf17 0xf18 0xf22 0xf27 0xf2f 0xf37 0xf59 0xf5f 0xf77 0xf88 0xfa7 0xfaf
0xfb1 0xfc9 0xfcb 0xfd9 0xdd 0xfe7 0xfe9 0xfeb 0xfed 0xff5 0xff6 0xffc

B.2.3 Preservación de patrones periódicos

La mayoría de los patrones periódicos se preservan mediante la regla general. Se necesitan algunas excepciones, no obstante, para obtener mejores prestaciones en las regiones de transición a y desde patrones periódicos. A continuación se indican las 10 excepciones de patrones periódicos.

0x638 0xa38 0x692 0xc92 0xaaa 0xcaa 0xb55 0xd55 0x36d 0x5c7

B.2.4 Preservación de patrones de tremolación

Las 12 excepciones de patrones de tremolación ayudan a preservar la tremolación de muy baja densidad o muy alta densidad, es decir, pixels blancos o negros aislados. Se indican a continuación:

0x010 0x028 0x082 0x085 0xeba 0xebd 0x142 0x145 0xf7a 0xf7d 0xfd7 0xfef

Anexo C

Cambios de plantilla adaptativa

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

C.1 Generalidades

La técnica descrita en este anexo es sencilla desde el punto de vista del cálculo y efectúa buenas determinaciones del momento en que es conveniente un cambio de pixel AT. Proporcionará una ganancia de codificación sustancial, aumentándola a veces por dos, en imágenes que contienen semitonos.

La descripción supone $M_Y = 0$, es decir, que sólo se autoriza el desplazamiento en línea del pixel AT. (El soporte mínimo sugerido es sólo para esta situación.) Es evidente una generalización para $M_Y \neq 0$, pero el algoritmo resultante no ha sido probado, y con M_Y grande es sustancial el procesamiento necesario por el codificador.

El algoritmo hace una comprobación al comienzo de cada barra para determinar cuál de los pixels AT tiene el máximo valor predictivo para el pixel-objetivo. Siempre que se compruebe que un pixel AT no actualmente configurado en la plantilla tiene un valor predictivo mucho mayor que el que sí lo está, podría ser conveniente una conmutación. Es esencial no obstante que cualquiera de estas conmutaciones de plantilla sólo se haga poco frecuentemente y por una razón poderosa. Siempre que se haga una conmutación de plantilla, las estimaciones de probabilidad mantenidas por el codificador aritmético se vuelven mediocres hasta que pasa tiempo suficiente para que se produzca readaptación.

C.2 Capas diferenciales

Las Figuras C.1 y C.2 presentan un diagrama de flujo para el procesamiento de AT en una capa diferencial. Se utiliza para medir el valor predictivo una formación de contadores. El contador c_n , $n = 0$ ó $3 \leq n \leq M_X$, cuenta el número de coincidencias de polaridad entre el pixel-objetivo y el pixel AT candidato n . El pixel AT candidato 0 es el pixel AT por defecto. Los pixels candidatos n , $3 \leq n \leq M_X$ son pixels a $\tau_X = n$ y $\tau_Y = 0$. Se dice que se ha producido una coincidencia de polaridad cuando ambos pixels son blancos o ambos son negros.

El diagrama de flujo de las Figuras C.1 y C.2 se ejecuta una vez al comienzo de cada barra. Los contadores se reponen a cero, y se cuentan las coincidencias hasta el final de alguna línea, siendo la máxima cuenta posible mayor que o igual a 2048. Cuando se llega a este número, se comprueban un nido de condiciones, y si todas ellas se satisfacen, se efectúa una conmutación de plantilla. En otro caso, se deja la plantilla como estaba al menos durante el resto de la barra.

Sólo la comprobación una vez por barra para una posible conmutación de plantilla es un modo razonable de ahorrar cálculos, si el número de barras por imagen no es apreciablemente menor que el número sugerido, 35. Si existen simplemente unas pocas barras por imagen, o especialmente si sólo existe una barra por imagen, se necesita comprobación continua o periódica dentro de cada barra.

A resoluciones muy bajas puede no haber 2048 pixels en una barra. En este caso, no se harán con este algoritmo cambios de plantilla AT. Dicho comportamiento es probablemente deseable. Con demasiados pocos pixels, los datos son ruidosos y no pueden proporcionar orientación fiable sobre el lugar adonde ha de desplazarse el pixel AT.

Si se ha determinado que es deseable una conmutación AT, hay dos selecciones razonables sobre el momento en que ha de hacerse efectiva. Lo más sencillo es hacer efectiva la conmutación al comienzo de la próxima línea. Este planteamiento ofrece la máxima ganancia de codificación, pero exige que el procesamiento AT se realice en un pase previo o que los SCD para una determinada barra se someta a memorización intermedia por un codificador de manera que permitan ser precedidos por un código marcador ATMOVE si se considera conveniente una conmutación AT mientras se codifica esa barra. Otra posibilidad, con muy pequeña pérdida de la eficacia de codificación (suponiendo de nuevo un número razonable de barras por imagen), es que la conmutación puede hacerse efectiva al comienzo de la siguiente barra $y_{AT} = 0$. En este caso, no es necesario que el código marcador AT aparezca hasta el comienzo de esa barra y es de nuevo posible alimentar el codificador.

C.3 Capa de resolución mínima

El procesamiento AT en la capa de resolución mínima puede hacerse casi idénticamente al descrito en la subcláusula anterior para capas diferenciales. Sólo hay que hacer tres cambios en la Figura C.1, el primero es suprimir la condición $DPVALUE = 2$ del bloque situado en medio de la Figura C.1, el segundo es sustituir la condición $x \geq M_X$ por $M_X \leq x < X_d - 2$, y el último cambiar apropiadamente el conjunto sobre el cual varía el índice contador n .

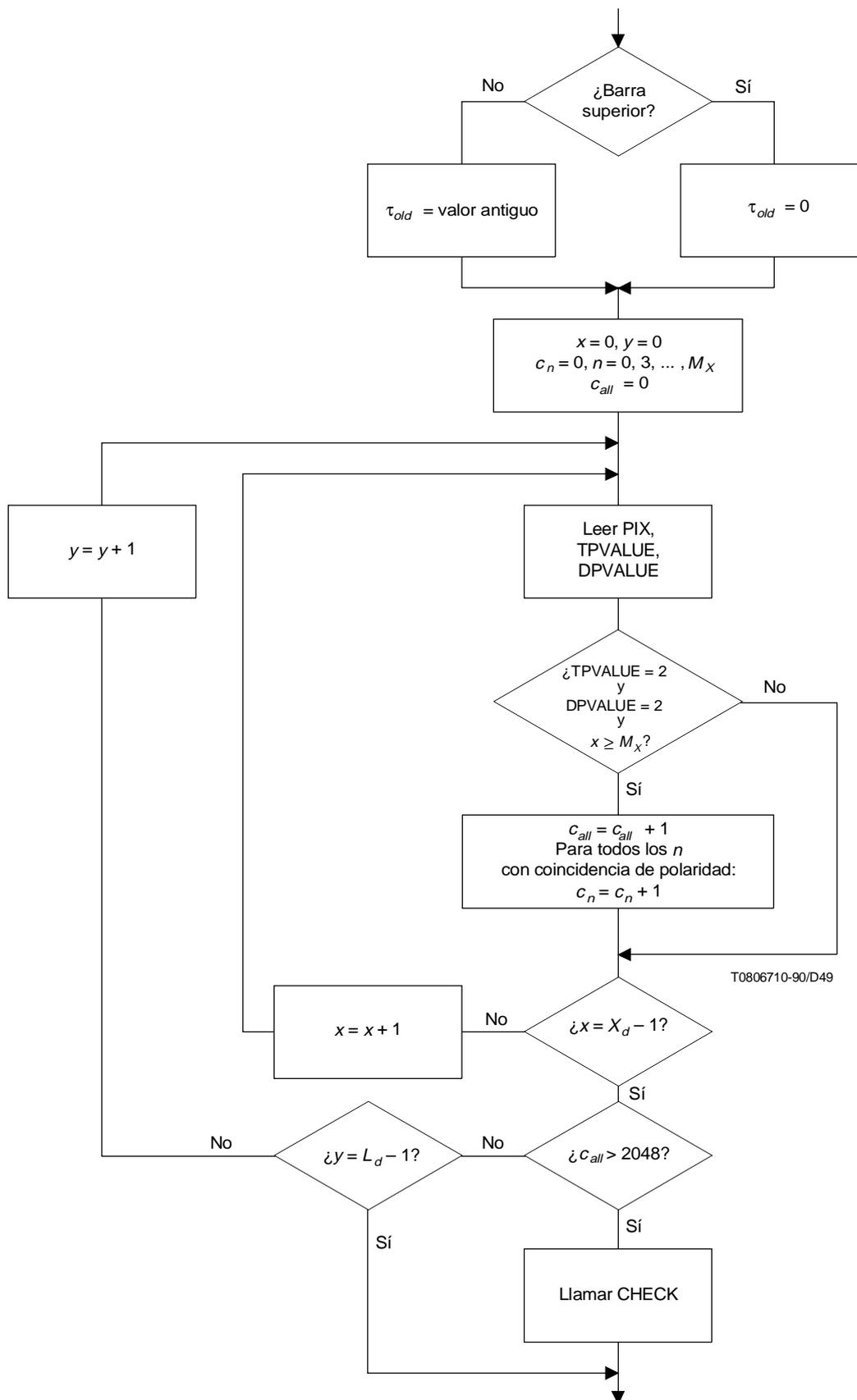


Figura C.1 – Diagrama de flujo para AT

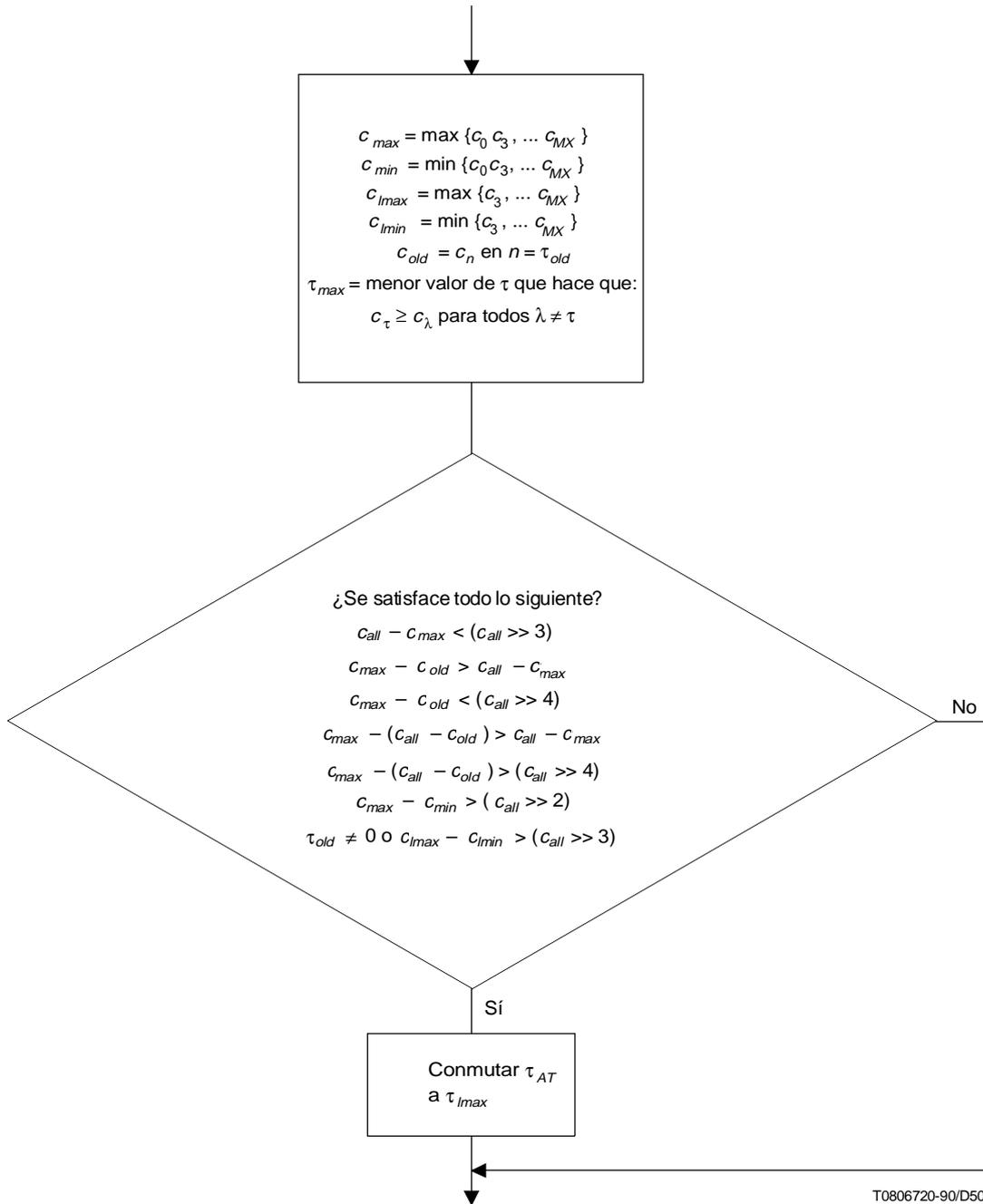


Figura C.2 – Diagrama de flujo para el procedimiento CHECK

Anexo D

Diseño de la tabla de estimación de probabilidades

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

D.1 Estimación bayesiana

Sea x_0, x_1, \dots una secuencia de variables aleatorias binarias independientes, idénticamente distribuidas, que adoptan los valores 1 y 0 con probabilidades p_1 y $p_0 = (1 - p_1)$. Se designa por $n_1(k)$ el número de unos en la secuencia x_0, x_1, \dots, x_{k-1} , y por $n_0(k)$ el número de ceros de esa misma secuencia. (De aquí que $n_0(k) + n_1(k) = k$). Si p_1 es el mismo una variable aleatoria con una distribución uniforme en $[0,1]$, entonces la estimación bayesiana $\hat{p}_1(k)$ de p_1 , dada una observación x_0, x_1, \dots, x_{k-1} viene dada por:

$$\hat{p}_1(k) = \frac{n_1(k) + 1}{n_1(k) + 1 + n_0(k) + 1} \quad (\text{D-1})$$

La estimación:

$$\hat{p}_1(k) = \frac{n_1(k) + \delta}{n_1(k) + \delta + n_0(k) + \delta} \quad (\text{D-2})$$

con $\delta \in (0,1)$ es también una estimación bayesiana. Ahora bien, la distribución determinada *a priori* para la que la estimación es bayesiana es una que hace que los valores de p_1 próximos a 0 y 1 tengan una probabilidad mayor que la de los valores próximos a $\frac{1}{2}$. Mientras menor es el valor de δ , más sesgada es la probabilidad hacia 0 y 1.

D.2 Múltiples contextos

El codificador aritmético de esta especificación opera en un entorno de múltiples contextos. Al codificar $\text{PIX}(k)$, se suministra con un contexto $\text{CX}(k)$. Una estimación bayesiana de la probabilidad $p_1(k)$ de que $\text{PIX}(k)$ sea igual a 1 viene dada por:

donde δ es también esta vez un parámetro libre en $(0,1)$, $n_{1,\text{CX}}(k)$ designa el número de veces que $\text{PIX}(i)$, $i \in [0, k-1]$, ha sido 1 en el contexto $\text{CX}(k)$, y $n_{0,\text{CX}}(k)$ se define análogamente para 0.

$$\hat{p}_1(k) = \frac{n_{1,\text{CX}}(k) + \delta}{n_{1,\text{CX}}(k) + \delta + n_{0,\text{CX}}(k) + \delta} \quad (\text{D-3})$$

Para más facilidad de presentación en el resto de este anexo, se supondrá un entorno de contexto simple, pero todos los conceptos y fórmulas que han de desarrollarse son elementalmente generalizables para múltiples contextos del mismo modo que la ecuación (D-3) generaliza la ecuación (D-2).

D.3 Parametrización MPS/LPS

Es conveniente reparametrizar la ecuación (D-2). Hagamos $\text{MPS}(k)$ igual a 1 ó 0, ya que hay respectivamente más unos o ceros en la secuencia x_0, x_1, \dots, x_{k-1} . (Si están en igual número, $\text{MPS}(k)$ puede definirse como 0 ó 1.) Entonces

$$\hat{p}_1(k) = \begin{cases} \hat{p}_{LPS}(k), & \text{si } \text{MPS}(k) = 0 \\ 1 - \hat{p}_{LPS}(k), & \text{si } \text{MPS}(k) = 1 \end{cases} \quad (\text{D-4})$$

donde

$$\hat{p}_{LPS}(k) = \frac{n_{LPS}(k) + \delta}{n_{MPS}(k) + \delta + n_{LPS}(k) + \delta} \quad (\text{D-5})$$

y $n_{LPS}(k)$ y $n_{MPS}(k)$ son cuentas para, respectivamente, los símbolos más probable y menos probable de x_0, x_1, \dots, x_{k-1} .

Las iteraciones para $n_{LPS}(k)$, $n_{MPS}(k)$, y $MPS(k)$ vienen dadas por

$$n_{LPS}(k+1) = \begin{cases} n_{LPS}(k) + 1, & \text{si } x_k \neq MPS(k) \text{ y } n_{LPS}(k) \neq n_{MPS}(k) \\ n_{LPS}(k), & \text{en otro caso} \end{cases} \quad (D-6)$$

$$n_{MPS}(k+1) = \begin{cases} n_{MPS}(k) + 1, & \text{si } x_k = MPS(k) \text{ o } n_{LPS}(k) = n_{MPS}(k) \\ n_{MPS}(k), & \text{en otro caso} \end{cases} \quad (D-7)$$

y

$$MPS(k+1) = \begin{cases} 1 - MPS(k), & \text{si } x_k \neq MPS(k) \text{ y } n_{LPS}(k) = n_{MPS}(k) \\ MPS(k), & \text{en otro caso} \end{cases} \quad (D-8)$$

D.4 Seguimiento rápido

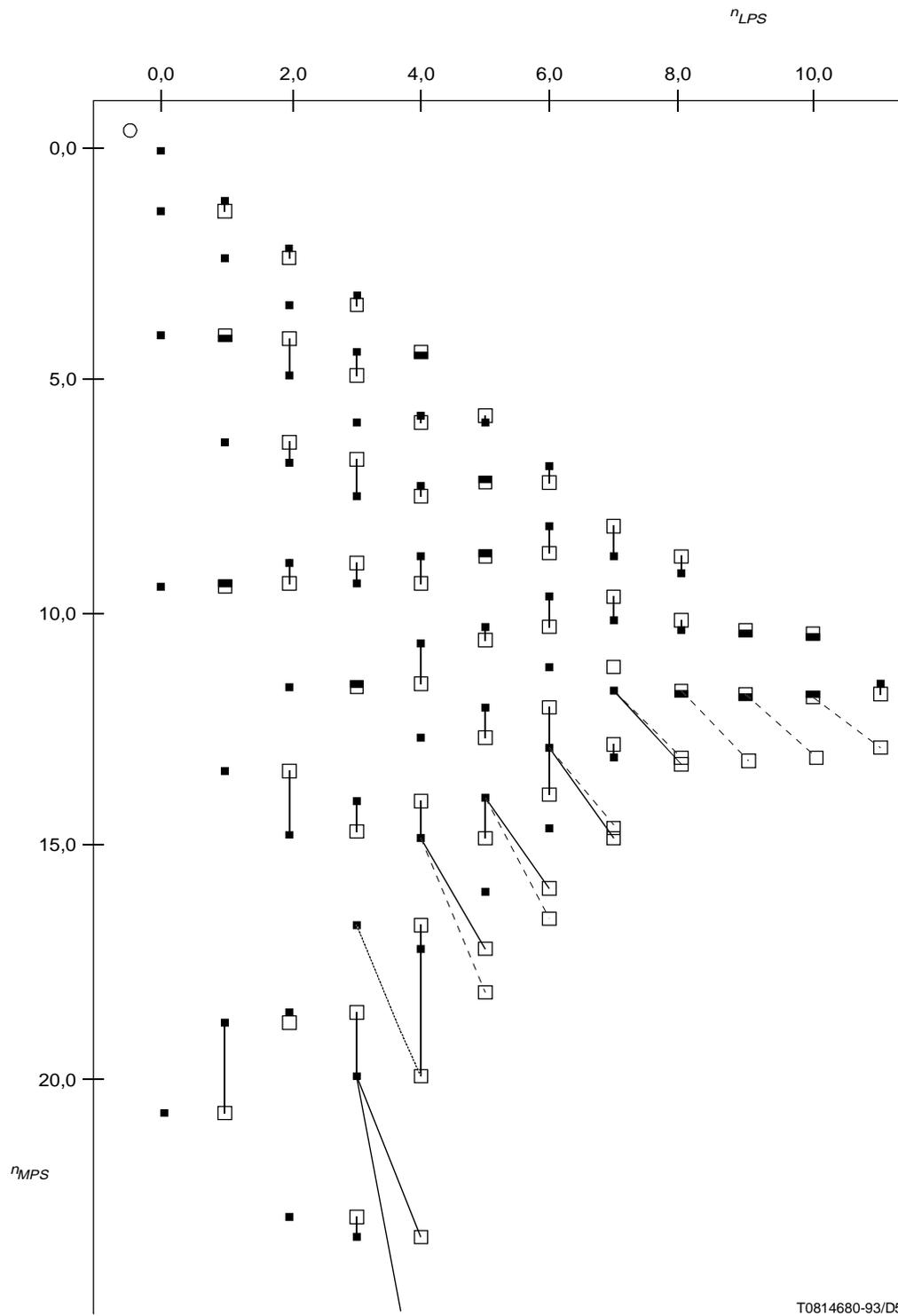
La estimación de p_{LPS} dada por (D-5) es una estimación bayesiana y una excelente estimación si, como se supone en su obtención, la secuencia $x_0, x_1 \dots$ es estacionaria. Sin embargo, en la aplicación de la codificación aritmética de esta Especificación, la entrada podría no ser estacionaria y puede cambiar su naturaleza estadística a medida que se codifican diferentes porciones de una imagen. Para obtener una buena eficacia de codificación, es importante que el estimador de probabilidad siga los cambios en los datos estadísticos de entrada. El problema con la estimación (D-5) de p_{LPS} en un entorno no estacionario es que se «atasca». Una vez que n_{LPS} y n_{MPS} se han elevado a grandes números, se necesitan muchas observaciones contrarias para cambiar apreciablemente p_{LPS} .

Al mismo tiempo, como es natural, se desea también una estimación exacta (no ruidosa) en el entorno de régimen permanente. Un excelente compromiso entre el seguimiento rápido y la estimación exacta en régimen permanente puede obtenerse fijando n_{LPS} . Cuando la iteración (D-6), (D-7), (D-8) hace que n_{LPS} supere algún umbral, entonces tanto éste como n_{MPS} se someten a un escalamiento regresivo proporcional. Como el escalamiento es proporcionado, no afecta a p_{LPS} . Sin embargo, mantiene n_{LPS} y n_{MPS} pequeños, a fin de que sea rápida la respuesta a los cambios de las estadísticas fundamentales. La fijación exacta del umbral de fijación para desencadenar un escalamiento proporcionado permite establecer un compromiso entre seguimiento rápido y exactitud de estimación. Pequeños umbrales favorecen un seguimiento rápido y grandes umbrales favorecen la exactitud de estimación.

D.5 Reducción de la carga de cálculo

El Cuadro 24 define en realidad un estimador de probabilidad. Este estimador imita la versión con fijación de la iteración (D-6), (D-7), (D-8) recién tratada. Sin embargo, es importante decir que esta imitación se efectúa de modo que minimice la carga de cálculo.

Las Figuras D.1A y D.1B muestran la misma representación, pero a dos resoluciones verticales distintas que difieren en un factor de 1000. Esta representación muestra gráficamente, en el espacio $n_{LPS} - n_{MPS}$, los datos que se presentan tabularmente en el Cuadro 24. Hay 113 cuadrados de color uniforme en la representación y cada uno de los cuales corresponde a uno de los estados mostrados en el Cuadro 24. Cada uno de estos estados tiene una estimación de probabilidad de \hat{p}_{LPS} asociada con el mismo mediante la ecuación (D-5). Se eligió un valor de δ de 0,45 por optimización experimental. Para mayor comodidad gráfica posterior, el punto $(-\delta, -\delta)$ se muestra como un círculo en las Figuras D.1A y D.1B. Todas las líneas que radian desde este punto son líneas de probabilidad constante.



T0814680-93/D51

Figura D.1A – Estados del estimador de probabilidad en el espacio $n_{LPS} - n_{MPS}$:
 Máximo valor de n_{MPS} representado = 25

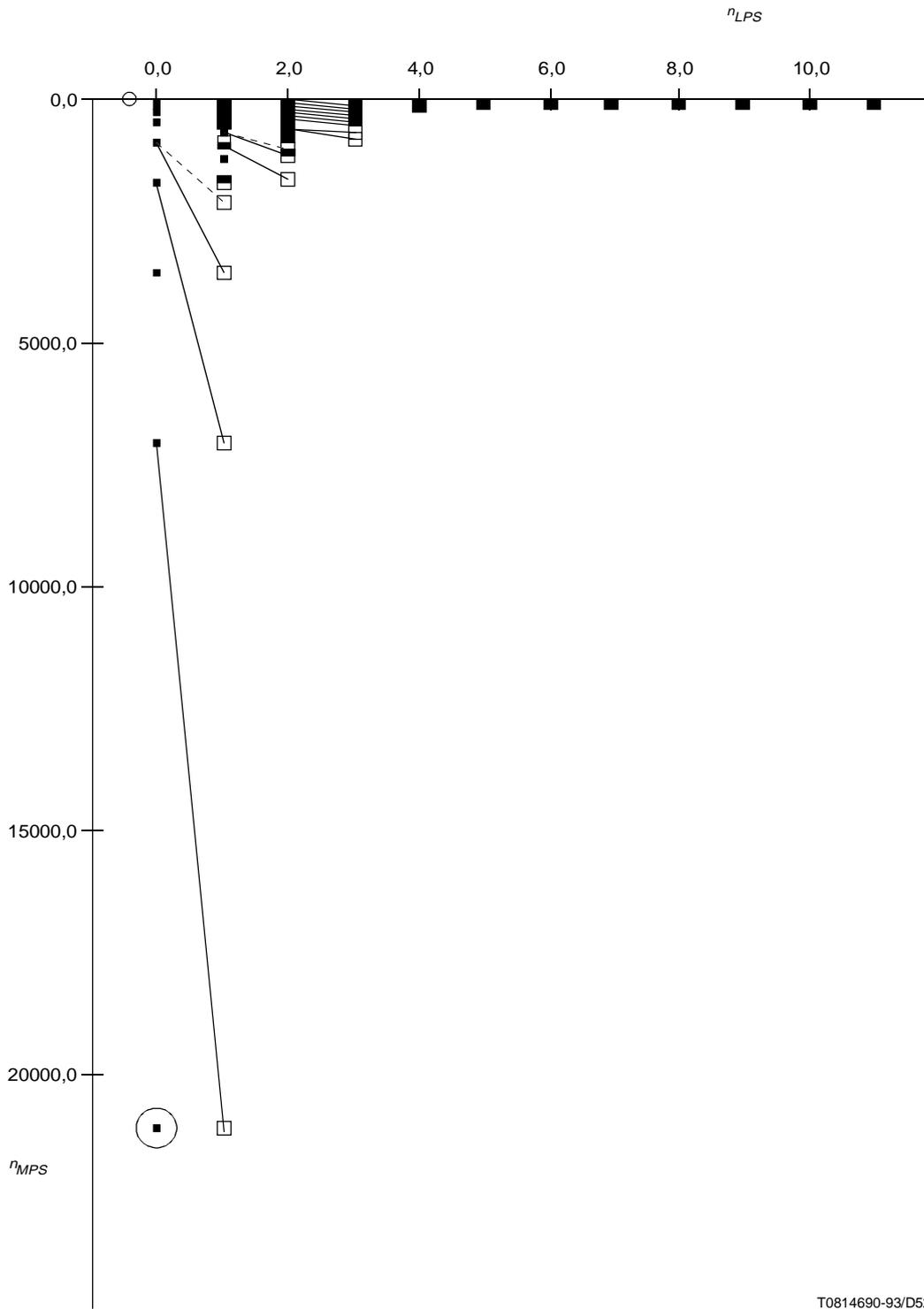


Figura D.1B – Estados del estimador de probabilidad en el espacio $n_{LPS} - n_{MPS}$:
 Máximo valor de n_{MPS} representado = 25 000

La columna designada por L_{SZ} en el Cuadro 24 se obtiene en primera aproximación utilizando la ecuación 8 (véase 6.8.1.2). El intervalo de codificación A tiene una densidad de probabilidad que se determina con buena aproximación haciéndola inversamente proporcional a A de modo que \bar{A} en la ecuación (8) (véase 6.8.1.2) sea:

$$0,721 \times 0x10000 = 0xb893 \quad (D-9)$$

Las entradas efectivas del Cuadro 24 difieren de $p_{LPS} \times 0xb893$ en un pequeño porcentaje debido a que se ha efectuado cierta optimización experimental adicional.

Cuando se recibe MPS, n_{MPS} debe idealmente incrementarse en 1 como en la ecuación (D-7). Sin embargo, dicho procedimiento conduciría a un número de estados irrazonablemente grande. En su lugar el desplazamiento hacia abajo en la representación (D-1) está condicionado a que exista una renormalización. Designemos por:

$$a = A / 0x10000 \quad (D-10)$$

el tamaño de registro A . La probabilidad P_{RN} de una renormalización provocada por MPS viene dada por:

$$\begin{aligned} P_{RN} &= P_R \{ (1-p)a < (1/2) \} \\ &= P_R \{ a < (1/2)/(1-p) \} \\ &= \log_2(1/(1-p)) \end{aligned} \quad (D-11)$$

donde se ha supuesto de nuevo que la variable aleatoria a tiene una densidad en $(1/2, 1)$ inversamente proporcional a a . En cualquier columna vertical la distancia entre cualesquiera dos estados es igual a $1/P_{RN}$, donde P_{RN} es el indicado por (D-11) con p igual a p_{LPS} para el más alto de los dos estados. Este gran incremento compensa exactamente el hecho de que n_{MPS} esté cambiándose sólo con una probabilidad P_{RN} .

Cuando el estado vigente está en el fondo de su columna, es por supuesto imposible descender en esa columna por debajo de $1/P_{RN}$. En la representación (D-1) tales estados tienen inmediatamente por debajo de ellos una marca cuadrada abierta en la que la actualización de MPS desearía provocar n_{MPS} . El punto deseado está conectado por una línea de trazo interrumpido al estado al que se ha desplazado realmente. Obsérvese que las líneas de trazo interrumpido radian todas hacia el círculo en $(-\delta, -\delta)$, y por tanto el remapeamiento cambia p_{LPS} lo menos posible.

Cuando se recibe el LPS hay siempre una renormalización y el desplazamiento nominal en el espacio $n_{LPS} - n_{MPS}$ pasa a $(n_{LPS} + 1 - n_{MPS})$. Tales puntos se conocen como cuadrados abiertos en la representación (D-1). En todos los casos estos cuadrados abiertos no coinciden con ningún estado disponible y deben remapearse. Cuando la actualización nominal es «interna» al área del espacio $n_{LPS} - n_{MPS}$ cubierto por los estados disponibles, el remapeamiento se efectúa por un desplazamiento vertical al estado disponible más próximo (en un sentido p_{LPS}). Estos mapeamientos se muestran mediante líneas de trazo continuo. Cuando la actualización nominal es «externa», es deseable la fijación descrita en D.4, y el remapeamiento se produce a un estado disponible de la columna uno a la izquierda del punto nominal. Una vez más el punto concreto elegido en esta columna es el más próximo al punto nominal en un sentido p_{LPS} . Los remapeamientos externos como éste también se muestran con líneas de trazo continuo en la representación (D-1).

Hay un estado que se retromapea en sí mismo al recibir un MPS. Se muestra con un círculo en torno al mismo y sólo aparece con la gran escala de la Figura D.1B. Este estado tiene la mínima probabilidad asociada de cualquiera de los estados (aproximadamente 0,00002) y no hay mejor estado al que hacer la transición al recibir el MPS.

El umbral de fijación efectivo es una función débil de n_{MPS} y por ende de p_{LPS} . Los valores de p_{LPS} próximos a $1/2$ se estiman con menos ruido, pero por ende con menos posibilidad de seguimiento que los valores próximos a 0. Se vio que este comportamiento era deseable al diseñar un codificador aritmético común para esta Especificación y también para la Rec. T.81 del CCITT | ISO/CEI 10918-1. Para la aplicación de esta Especificación, el seguimiento rápido es de relativamente más importancia, mientras que en la Rec. T.81 del CCITT | ISO/CEI 10918-1 la calidad de estimación es de relativamente más importancia. Es conveniente que los valores de p_{LPS} encontrados en la aplicación de esta especificación sean mucho menores que los encontrados en la aplicación de la Rec. T.81 del CCITT | ISO/CEI 10918-1. Hacer que el umbral de fijación resulte algo menor para pequeños p_{LPS} proporciona en efecto automáticamente el comportamiento deseado en cada una de las dos Especificaciones diferentes.

Anexo E

Patentes

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

E.1 Observaciones de introducción

Se señala a la atención del usuario que, a los efectos de la conformidad con esta Especificación, pueda ser necesario utilizar un invento protegido por derechos de patente.

Al publicar esta Especificación no se adopta ninguna posición con respecto a la validez de cualesquiera títulos o derechos de patente relacionados con ella. Sin embargo, para todas las patentes citadas en este anexo, el detentor de la patente ha entregado al ITTF (ISO/CEI *information technology task force*) y a la Oficina de Normalización de las Telecomunicaciones (TSB) una declaración en la que se compromete a otorgar licencia en términos y condiciones razonables y no discriminatorios a quienes deseen obtenerla.

Los criterios de inclusión de patentes en este anexo son los siguientes:

- 1) La patente ha sido identificada por alguna persona familiarizada con los campos técnicos relevantes para esta Especificación, que considera que se requiere utilizar el invento cubierto por la patente para la realización de uno o varios de los procesos de codificación especificados.
- 2) El detentor de la patente ha enviado una carta al ITTF y al TSB expresando su disposición a otorgar licencia a un número ilimitado de solicitantes en todo el mundo en términos razonables y en condiciones que pueden demostrarse estar libres de toda discriminación injusta.

La lista de patentes se actualizará durante la preparación y el mantenimiento de esta Especificación si es necesario, con ocasión de las revisiones de la Recomendación | Norma Internacional.

E.2 Lista de patentes

Se enumeran únicamente las patentes en los países de las empresas detentoras. En muchos casos se han registrado patentes en el extranjero.

- 1) IBM, *A method and means for pipeline decoding of the high to low order pairwise combined digits of a decodable set of relatively shifted finite number of strings*, US 4 295 125, 13 de octubre de 1981.
- 2) IBM, *A method and means for carry-over control in a high order to low order combining of digits of a decodable set of relatively shifted finite number strings*, US 4 463 342, 31 de julio de 1984.
- 3) IBM, *High-speed arithmetic compression using concurrent value updating*, US 4 467 317, 21 de agosto de 1984.
- 4) IBM, *Method and means for arithmetic coding using a reduced number of operations*, US 4 286 256, 25 de agosto de 1981.
- 5) IBM, *A multiplication-free multi-alphabet arithmetic code*, US 4 652 856, 4 de febrero de 1986.
- 6) IBM, *Symmetrical adaptive data compression/decompression system*, US 4 633 490, 30 de diciembre de 1986.
- 7) IBM, *Arithmetic coding data compression/decompression by selectively employed, diverse arithmetic encoders and decoders*, US 4 891 643, 2 de enero de 1990.
- 8) IBM, *System for compression bi-level data*, US 4 901 363, 13 de febrero de 1990.
- 9) IBM, *Arithmetic coding encoder and decoder system*, US 4 905 297, 27 de febrero de 1990.
- 10) IBM, *Probability adaptation for arithmetic coders*, US 4 935 882, 19 de junio de 1990.

ISO/CEI 11544:1993(S)

- 11) IBM, *Probability adaptation for arithmetic coders*, US 5 099 440, 24 de marzo de 1992.
- 12) IBM, *Method and apparatus for processing pel signals of an image*, US 4 982 292, 1 de enero de 1991.
- 13) AT&T, *Progressive transmission of high resolution two-tone facsimile images*, US 4 870 497, 26 de septiembre de 1989.
- 14) AT&T, *Edge decomposition for the transmission of high resolution facsimile images*, US 4 873 577, 10 de octubre de 1989.
- 15) AT&T, *Adaptive probability estimator for entropy encoder/decoder*, US 5 025 258, 18 de junio de 1991.
- 16) AT&T, *Efficient encoding/decoding in the decomposition and recomposition of a high resolution image utilizing its low resolution replica*, US 4 979 049, 18 de diciembre de 1990.
- 17) AT&T, *Efficient encoding/decoding in the decomposition and recomposition of a high resolution image utilizing pixel clusters*, US 5 031 053, 9 de julio de 1991.
- 18) AT&T, *Entropy encoder/decoder including a context extractor*, US 5 023 611, 11 de junio de 1991.
- 19) AT&T, *Method and apparatus for carry-over control in arithmetic entropy coding*, US 4 973 961, 27 de noviembre de 1990.
- 20) KDD, *Methods for reduced-sized images*, Japan Application No. 63-212 432, pendiente en Japón.
- 21) KDD, *Image reduction system*, Japan Application No. 1-167 033, conjunta con Canon, pendiente en Japón.
- 22) Mitsubishi, *Facsimile encoding communication system*, Japan 1 251 403, 6 de julio de 1984.
- 23) Mitsubishi, *Encoding method*, pendiente en Japón.
- 24) Canon, *Image reduction system*, Japan Application No. 1-167 033, conjunta con KDD, pendiente en Japón.

E.3 Direcciones de contacto para informaciones sobre patentes

Director del CCITT

Unión Internacional de Telecomunicaciones

Place des Nations

CH-1211 Genève 20

Suiza

Tel: +41 (22) 730 5111

Fax: +41 (22) 730 5853

ITTF

International Organization for Standardization

1, rue de Varembe

CH-1211 Genève 20

Suiza

Tel: +41 (22) 734 0150

Fax: +41 (22) 733 3843

Program Manager, Licensing
Intellectual Property and Licensing Services
IBM Corporation
208 Harbor Drive
P.O. Box 10501
Stamford, Connecticut 08904-2501
Tel: +1 (203) 973 7935
Fax: +1 (203) 973 7981 ó +1 (203) 973 7982

Mitsubishi Electric Corp.
Intellectual Property License Department
1-2-3 Morunouchi, Chiyoda-ku
Tokyo 100, Japan
Tel: +81 (3) 3218 3465
Fax: +81 (3) 3215 3842

International Affairs Department
Kokusai Denshin Denwa Co. Ltd.
3-2 Nishishinjuku 2-chome
Shinjuku-ku
Tokyo 163, Japan
Tel: +81 (3) 3347 6457
Tel: +81 (3) 3347 6470

AT&T Intellectual Property Division Manager
Room 3A21
10 Independence Blvd.
Warren, NJ 07059, USA
Tel: +1 (908) 580 5392
Fax: +1 (908) 580 6355

Senior General Manager
Corporate Intellectual Property and Legal Headquarters
Canon Inc.
30-2 Shimomaruko 3-chome
Ohta-ku, Tokyo 146, Japan
Tel: +81 (3) 3758 2111
Fax: +81 (3) 3756 0947

Anexo F

Bibliografía

(Este anexo no es parte integrante de esta Recomendación | Norma Internacional)

ISO/CEI 9281-1, *Information technology – Picture coding methods – Part 1: Identification*.

Recomendación T.81 del CCITT (1992) | ISO/CEI 10918-1:1993, *Tecnología de la información – Compresión digital y codificación de imágenes fijas de tonos continuos – Requisitos y directrices*.

Recomendación T.4 del CCITT, *Normalización de los aparatos facsímil del grupo 3 para la transmisión de documentos*.

Recomendación T.6 del CCITT, *Esquemas de codificación facsímil y funciones de control de codificación para los aparatos facsímil del grupo 4*.

WITTEN (I. H.), NEAL (R. M.) y CLEARY (J. G.): Arithmetic coding for data compression, *Communications of the ACM*, Vol. 30, No. 6, pp. 520-540, junio de 1987.

ARPS (R. B.), TRUONG (T. K.), LU (D. J.), PASCO (R. C.) y FRIEDMAN (T. D.): A multi-purpose VLSI chip for adaptive data compression of bilevel images, *IBM Journal of Research and Development*, Vol. 32, No. 6, pp. 775-795, noviembre de 1988.

PENNEBAKER (W. B.), MITCHELL (J. L.), LANGDON (G. G.) Jr. y ARPS (R. B.): An overview of the basic principles of the Q-Coder adaptive binary arithmetic type coder, *IBM Journal of Research and Development*, Vol. 32, No. 6, pp. 717-726, noviembre de 1988.

ONO (F.), KINO (S.), YOSHIDA (M.) y KIMURA (T.): Bi-level image coding with Melcode – Comparison of block type code and arithmetic type code, *IEEE Global Telecommunications Conference*, pp. 255-260, noviembre de 1989.

DUTTWEILER (D.) y CHAMZAS (C.): Probability estimation in arithmetic and adaptive-Huffman entropy coders, se publicará en *IEEE Trans. on Image Processing*.

SHEINWALD (D.) y PASCO (R.): Deriving deterministic prediction rules from reduction schemes, se publicará en *IEEE Trans. Information Theory*.

LANGDON (G.): Method for carry-over control in a FIFO arithmetic code string, *IBM Technical Disclosure Bulletin*, Vol. 23, No. 1, pp. 310-312, junio de 1980.

JONES (C.): An efficient coding system for long source sequences, *IEEE Trans. Information Theory*, Vol. IT-27, pp. 280-291, mayo de 1981.

LANGDON (G.): An introduction to arithmetic coding, *IBM Journal of Research and Development*, Vol. 28, pp. 135-149, marzo de 1984.