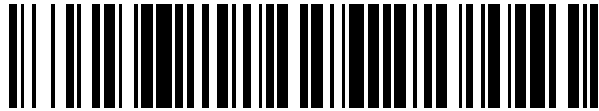


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 442 215**

51 Int. Cl.:

H04N 7/26

(2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **02.05.2003 E 08020010 (8)**

97 Fecha y número de publicación de la concesión europea: **06.11.2013 EP 2037412**

54 Título: **Procedimiento y dispositivo de codificación y decodificación aritméticas de estados binarios y programa informático correspondiente y soporte de almacenamiento correspondiente legible por ordenador**

30 Prioridad:

02.05.2002 DE 10220962

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

10.02.2014

73 Titular/es:

**FRAUNHOFER-GESELLSCHAFT ZUR
FÖRDERUNG DER ANGEWANDTEN
FORSCHUNG E.V. (100.0%)
Hansastraße 27c
80686 München, DE**

72 Inventor/es:

**MARPE, DETLEF y
WIEGAND, THOMAS**

74 Agente/Representante:

PONTI SALES, Adelaida

ES 2 442 215 T3

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

DESCRIPCIÓN

Procedimiento y dispositivo de codificación y descodificación aritméticas de estados binarios y programa informático correspondiente y soporte de almacenamiento correspondiente legible por ordenador

5 **[0001]** La invención se refiere a un procedimiento y a un dispositivo para codificar aritméticamente y descodificar estados binarios y a un programa informático correspondiente y un medio de almacenamiento legible por ordenador correspondiente que puede en particular ser utilizado en la compresión de datos digitales.

10 **[0002]** La presente invención describe un nuevo procedimiento eficiente para la codificación aritmética binaria. Existe una demanda para la codificación aritmética binaria en la mayoría de áreas de aplicación diferentes de compresión de datos digitales; aquí, en particular, son especialmente interesantes las aplicaciones en los sectores de la compresión de imágenes digitales. En numerosas normas para la codificación de imágenes, como por ejemplo, JPEG, JPEG -2000, JPEG -LS y JPIG, se definieron procedimientos para una codificación aritmética binaria. Las actividades de normalización más nuevas también prevén de manera obvia el uso futuro de estas tecnologías de codificación en el campo de la codificación video (CABAC en H. 264/AVC) [1].

15 **[0003]** Las ventajas de la codificación aritmética (CA) en contraste con la codificación de Huffman [2] que hasta ahora se había utilizado en la práctica, se pueden caracterizar básicamente por tres características:

20 1. Mediante el uso de la codificación aritmética, por mecanismos de adaptación simples se puede obtener una adaptación dinámica a la presente estadística de fuente (adaptabilidad).

25 2. La codificación aritmética permite la asignación de un número de bits no entero por símbolo que va a codificarse y es por lo tanto adecuada para lograr resultados de codificación que ilustran una aproximación de la entropía como límite inferior dado teóricamente (aproximación de entropía) [3].

30 3. Se pueden utilizar enlaces estadísticas de modelos de contexto adecuados entre símbolos para una reducción adicional de datos con la codificación aritmética (redundancia entre símbolos) [4].

[0004] Como desventaja de una aplicación de la codificación aritmética, se considera en general el aumento del esfuerzo de cálculo en comparación con la codificación de Huffman.

35 **[0005]** El concepto de la codificación aritmética se remonta a la documentación básica para la teoría de información por Shannon [5]. Los primeros procedimientos de construcción conceptuales fueron publicados primeramente por Elias [6]. Una primera variante LIFO (*last-in-first-out*) de la codificación aritmética fue diseñada por Rissanen [7] y más tarde modificada [8] [9] [10] por diferentes autores de las implementaciones FIFO (*first-in-first-out*).

40 **[0006]** Todos estos documentos tienen como principio básico común la descomposición de intervalo parcial recursiva. En correspondencia con las probabilidades dadas P("0") y P("1") de dos resultados {"0", "1"} de un alfabeto binario un intervalo dado principalmente, por ejemplo el intervalo [0, 1), se descompone de forma recursiva en intervalos parciales dependiendo de la ocurrencia de eventos individuales. Aquí, el tamaño del intervalo parcial resultante como el producto de las probabilidades individuales de los eventos que ocurren es proporcional a la probabilidad de la secuencia de eventos individuales. Como cada evento S_i añade una contribución de H(S_i) = -log(P(S_i)) del contenido de información teórica H(S_i) de S_i a la tasa global por la probabilidad P(S_i), una relación entre el número N_{Bit} de bits para ilustrar el intervalo parcial y la entropía de la secuencia de los resultados de los eventos individuales, que viene dada por el término derecho de la ecuación siguiente:

$$N_{Bit} = -\log \prod_i P(S_i) = -\sum_i \log P(S_i)$$

50 **[0007]** El principio básico, sin embargo, requiere en primer lugar de una precisión ilimitada (teóricamente) en la ilustración del intervalo parcial resultante y aparte de esto, tiene la desventaja de que sólo después de la codificación de los últimos resultados pueden darse como salida los bits para una representación de intervalo parcial resultante. A los efectos de aplicación práctica, fue decisivo por lo tanto desarrollar mecanismos para una salida gradual de bits con una representación simultánea con números con una precisión fija predeterminada. Estos mecanismos se introdujeron por primera vez en los documentos [3] [7] [11].

55 **[0008]** En la figura 1, se indican las operaciones básicas para una codificación aritmética binaria. En la implementación ilustrada el intervalo parcial actual está representado por los dos valores L y R, donde L indica el punto de desplazamiento y R el tamaño (anchura) del intervalo parcial, donde ambas cantidades se ilustran, utilizando números b bits enteros respectivamente. La codificación de un bit $\in \{0, 1\}$ se lleva a cabo de este modo, básicamente, en cinco subetapas: En la primera etapa se determina el valor del símbolo menos probable utilizando la estimación de probabilidad. Para este símbolo, también llamado LPS (*Least Probable Symbol*- símbolo menos probable), en contraste con el MPS (*Most Probable Symbol*- símbolo más probable), se utiliza la estimación de

probabilidad P_{LPS} en la segunda etapa para calcular la anchura R_{LPS} del intervalo parcial correspondiente. Dependiendo del valor del bit que hay que codificar L y R se actualizan en la tercera etapa. En la cuarta etapa se actualiza la estimación de probabilidad en función del valor del bit que se acaba de codificar y finalmente el intervalo de código R se somete a la así llamada renormalización en la última etapa, es decir R se reescala por ejemplo para que se satisfaga la condición $R \in [2^{b-2}, 2^{b-1}]$. Aquí, se da como salida un bit en cada operación de escalado. Para más detalles, consulte [10].

[0009] La principal desventaja de una implementación, tal como se indica más arriba, reside ahora en el hecho de que el cálculo de la anchura de intervalo R_{LPS} requiere una multiplicación para cada símbolo que hay que codificar. En general, las operaciones de multiplicación, en particular cuando se realizan en hardware, son costosas y requieren mucho tiempo. En varios documentos de investigación se examinaron los procedimientos para sustituir esta operación de multiplicación por una aproximación adecuada [11] [12] [13] [14]. Por la presente, los procedimientos publicados con referencia a este tema se pueden separar generalmente en tres categorías.

[0010] El primer grupo de propuestas para una codificación aritmética binaria libre de multiplicaciones, se basa en el enfoque de la aproximación de las probabilidades estimadas P_{LPS} para que la multiplicación en la segunda etapa de La figura 1 podrá sustituirse por una o varias operaciones de suma y desplazamiento [11] [14]. Para esto, en el caso más simple las probabilidades P_{LPS} se aproximan por valores en la forma 2^{-q} con el entero $q > 0$.

[0011] En la segunda categoría de procedimientos aproximativos se propone aproximar el rango de valores de R por valores discretos en la forma $(1/2 - r)$, donde se selecciona $r \in \{0\} \cup \{2^{-k} \mid k > 0, k \text{ entero}\}$ [15] [16].

[0012] La tercera categoría de procedimientos sólo se conoce por el hecho de que aquí cualquier operación aritmética se sustituye por accesos a tabla. La tercera categoría de procedimientos sólo se conoce por el hecho de que aquí cualquier operación aritmética se sustituye por accesos a tabla. A este grupo de procedimientos pertenecen, por un lado el codificador Q utilizado en los procedimientos JPEG estándar y los relacionados, tales como los codificadores QM - y MQ - [12], y por otro lado el codificador cuasi aritmético [13]. Mientras que este último procedimiento realiza una limitación drástica del número b de bits utilizados para la representación de R con el fin de obtener tablas aceptablemente dimensionadas, en el codificador Q la renormalización de R se implementa de manera que R puede al menos aproximadamente ser aproximado por 1. De esta manera se evita la multiplicación para determinar R_{LPS} . Además, se realiza la estimación de probabilidad mediante una tabla en la forma de una máquina de estado finito. Además, se realiza la estimación de probabilidad mediante una tabla en la forma de una máquina de estado finito [12].

[0013] El documento de patente US5592162 describe un procedimiento de realización de codificación aritmética, estando el número de posibles intervalos de anchuras restringido.

[0014] Mediante el uso de un índice relativo a dichas anchuras de intervalo, la codificación aritmética libre de multiplicaciones se puede lograr con una sobrecarga reducida en términos de almacenamiento.

[0015] Esta invención se refiere a un dispositivo y a un procedimiento para la codificación aritmética, un dispositivo y a un procedimiento para la decodificación aritmética de acuerdo con las reivindicaciones adjuntas.

[0016] Un procedimiento para codificar y decodificar estados binarios aritméticamente se realiza ventajosamente de modo que en una primera etapa un rango de valores que se puede pre-establecer para la especificación de la anchura de intervalo R se separa en K anchuras de intervalo representativas $\{Q_1, \dots, Q_K\}$, un rango de valores que se puede pre-establecer para la especificación de las probabilidades se separa en N estados de probabilidad representativos $\{P_1, \dots, P_N\}$ y se dan unas reglas de asignación, que asignan un Q_k ($1 \leq k \leq K$) a cada anchura de intervalo R y un P_n ($1 \leq n \leq N$) a cada probabilidad, y que en una segunda etapa la codificación o decodificación de los estados binarios ocurre realizando el cálculo de la nueva anchura de intervalo a obtener en el proceso de codificación o decodificación, respectivamente, utilizando una anchura de intervalo representativa Q_k ($1 \leq k \leq K$) y un estado de probabilidad representativo P_n ($1 \leq n \leq N$) mediante operaciones aritméticas distintas de la multiplicación y la división, en el que la anchura de intervalo representativa Q_k se determina mediante el intervalo de base básico de la anchura R y el estado de probabilidad representativo P_n se determina mediante la estimación de probabilidad subyacente al símbolo que hay que codificar o a decodificar según las reglas de asignación dadas.

[0017] Otra realización se caracteriza por el hecho de que basado en el intervalo que se tiene que evaluar actualmente con una anchura R para determinar la anchura de intervalo asociada Q_k se determina un índice q_index mediante una operación de desplazamiento y ocultamiento de bits aplicada a la representación interna/binaria en ordenador de R.

[0018] También es ventajoso que a partir del intervalo que se tiene que evaluar actualmente con una anchura R para determinar la anchura de intervalo asociada Q_k se determina un índice q_index mediante una operación de desplazamiento aplicada a la representación interna/binaria en ordenador de R y un acceso aguas abajo a una tabla

Qtab, en el que la tabla Qtab contiene los índices de anchuras de intervalo que corresponden a los valores de R que se precuantizaron por la operación de desplazamiento.

5 **[0019]** En particular, es ventajoso cuando la estimación de probabilidad subyacente al símbolo que hay que codificar o descodificar está asociada a un estado de probabilidad P_n utilizando un índice p_state .

10 **[0020]** También es una ventaja cuando la determinación de la anchura de intervalo R_{LPS} que corresponde a los LPS se realiza mediante un acceso a la tabla Rtab, en el que la tabla Rtab contiene los valores que corresponden a todos los K valores cuantizados de R y a los N diferentes estados de probabilidad de la anchura de intervalo R_{LPS} como valores de producto ($Q_K * P_n$). El esfuerzo de cálculo se reduce en particular cuando la determinación de la anchura de intervalo R_{LPS} que corresponde a los LPS se realiza mediante un acceso a la tabla Rtab, en el que para evaluar la tabla se utilizan el índice de cuantización q_index y el índice de la probabilidad $state\ p_state$.

15 **[0021]** También se prevé que en el procedimiento para los N diferentes estados de probabilidad representativos se den unas reglas de transición, en el que las reglas de transición indican que nuevo estado se utiliza basado en el símbolo codificado o descodificado actualmente para el siguiente símbolo que hay que codificar o descodificar. Por lo tanto es una ventaja cuando se crea una tabla *Next_State_LPS* que contiene el índice m del nuevo estado de probabilidad P_m cuando un símbolo menos probable (LPS) ocurre además del índice n del estado de probabilidad dado actualmente P_n , y/or cuando se crea una tabla *Next_State_MPS* que contiene el índice m del nuevo estado de probabilidad P_m cuando un símbolo más probable (MPS) ocurre además del índice n del estado de probabilidad dado actualmente P_n .

25 **[0022]** Se consigue una optimización del procedimiento para una codificación y decodificación aritmética binaria asistida mediante tabla, en particular, gracias a que los valores de la anchura de intervalo R_{LPS} que corresponden a todas las K anchuras de intervalo a a todos los N diferentes estados de probabilidad se entran como valores de producto ($Q_K * P_n$) en una tabla Rtab.

30 **[0023]** Una optimización adicional se consigue cuando el número K de los valores de cuantización y / o el número N de los estados representativos se seleccionan en función de la precisión preestablecida de la codificación y / o dependiendo del espacio de almacenamiento disponible.

[0024] La codificación incluye las siguientes etapas, por ejemplo:

- 35 1. Determinación del LPS
2. Cuantización de R:

$$q_index = Qtab[R \gg q]$$

- 40 3. Determinación de R_{LPS} y R:

$$R_{LPS} = Rtab [q_index, p_state]$$

$$R = R - R_{LPS}$$

- 45 4. Cálculo del nuevo intervalo parcial:

```

50   if (bit: LPS) then
       L ← L + R
       R ← RLPS
       p_state ← Next_State_LPS [p_state]
       if (p_state = 0) then valMPS ← 1 - valMPS
   else
       p_state ← Next_State_MPS [p_state]
```

- 55 5. Renormalización de L y R, bits de escritura, en el que

60 **q_index** describe el índice de un valor de cuantización leído de Qtab,
p_state describe el estado actual,
R_{LPS} describe la anchura de intervalo correspondiente al LPS y
valMPS describe el bit que corresponde al MPS.

[0025] La descodificación incluye las siguientes etapas, por ejemplo:

- 65 1. Determinación del LPS
2. Cuantización de R:

q_index = Qtab[R>>q]

3. Determinación de R_{LPS} y R:

5

$R_{LPS} = Rtab [q_index, p_state]$

$R = R - R_{LPS}$

4. Determinación de bit en función de la posición del intervalo parcial:

10

if ($V \geq R$) then

bit ← LPS

$V \leftarrow V - R$

15

$R \leftarrow R_{LPS}$

if ($p_state = 0$) valMPS ← 1 - valMPS

$p_state \leftarrow Next_State_LPS [p_state]$

else

bit ← MPS

20

$p_state \leftarrow Next_State_MPS [p_state]$

5. Renormalización de R, leer un bit y actualizar V, en el que

25

q_index

describe el índice de un valor de cuantización leído de Qtab,

p_state

describe el estado actual,

R_{LPS}

describe la anchura de intervalo correspondiente al LPS,

valMPS

describe el bit que corresponde al MPS, y

V

describe un valor del interior del intervalo parcial actual.

[0026] En otro procedimiento se proporciona que al codificar y/o descodificar el cálculo del índice de cuantización q_index se realiza en la segunda sub-etapa según la regla de cálculo:

30

$q_index = (R \gg q) \& Qmask$

35

en el que Qmask ilustra una máscara de bit seleccionada adecuadamente en función de K.

[0027] Si una distribución de probabilidad uniforme está presente se puede lograr una optimización adicional del procedimiento para una codificación descodificación aritmética asistida por tabla mediante la codificación según la reivindicación 12. Las sub-etapas 1 a 4 se realizan según la siguiente regla de cálculo:

40

$R \leftarrow R \gg 1$

if (bit = 1) then

$L \leftarrow L + R$

o donde las sub-etapas 1 a 4 de la codificación según la reivindicación 12 se realizan según la siguiente regla de cálculo:

45

$L \leftarrow L \ll 1$

if (bit = 1) then

$L \leftarrow L + R$

50

y en el que en la última alternativa la renormalización (subetapa 5 según la reivindicación 12) se realiza con valores umbral de decisión doblados y no doblados de L y R, y

donde en la descodificación según la reivindicación 13 las sub-etapas 1 a 4 se realizan según la siguiente regla de cálculo:

55

$R \leftarrow R \gg 1$

if ($V \geq R$) then

bit ← 1

$V \leftarrow V - R$

60

else

bit ← 0,

o las sub-etapas 1 a 5 de la descodificación según la reivindicación 13 se realizan según la siguiente regla de cálculo:

1. Leer un bit y actualizar V
2. Determinación de bit según la posición del intervalo parcial:

```

5           if (V ≥ R) then
              bit ← 1
              V ← V - R
            else
              bit ← 0.

```

10 **[0028]** También resulta ventajoso cuando la inicialización de los modelos de probabilidad se realiza en función de un parámetro de cuantización SliceQP y parámetros de modelo preestablecidos m y n, en el que SliceQP describe los parámetros de cuantización preestablecido al principio de una porción y m y n describen los parámetros de modelo.

15 **[0029]** También es ventajoso cuando la inicialización de los modelos de probabilidad incluye las siguientes etapas:

```

                1. preState = min(max(1, ((m * SliceQP) >>4)+n), 2*N)
                2. if (preState <=N) then
                    p_state = N+1 - preState
                    valMPS = 0
            else
                p_state = preState - N
                valMPS = 1,

```

25 en el que valMPS describe el bit que corresponde al MPS, SliceQP describe los parámetros de cuantización preestablecidos al principio de una porción y m y n describen los parámetros de modelo.

30 **[0030]** Un dispositivo para codificar y descodificar aritméticamente estados binarios incluye al menos un procesador que está implementado de manera que se pueda llevar a cabo un procedimiento para codificar y descodificar aritméticamente, en el que en una primera etapa un rango de valores que se puede pre-establecer para la especificación de la anchura de intervalo R se separa en K anchuras de intervalo representativas $\{Q_1, \dots, Q_K\}$, un rango de valores que se puede pre-establecer para la especificación de las probabilidades se separa en N estados de probabilidad representativos $\{P_1, \dots, P_N\}$ y se dan reglas de asignación, que asignan un Q_k ($1 \leq k \leq K$) a cada anchura de intervalo R y un P_n ($1 \leq n \leq N$) a cada probabilidad, y en el que en una segunda etapa la codificación o descodificación de los estados binarios ocurre realizando el cálculo de la nueva anchura de intervalo a obtener en el proceso de codificación o descodificación, respectivamente, utilizando una anchura de intervalo representativa Q_k ($1 \leq k \leq K$) y un estado de probabilidad representativo P_n ($1 \leq n \leq N$) mediante operaciones aritméticas distintas de la multiplicación y la división, en el que la anchura de intervalo representativa Q_k se determina mediante el intervalo de base básico de la anchura R y el estado de probabilidad representativo P_n se determina mediante la estimación de probabilidad subyacente al símbolo que hay que codificar o a descodificar según las reglas de asignación dadas..

45 **[0031]** Un programa de ordenador para codificar y descodificar aritméticamente de estados binarios permite a un ordenador, tras haber sido cargado en la memoria del ordenador, realizar un procedimiento para codificar y descodificar aritméticamente, en el que en una primera etapa un rango de valores que se puede pre-establecer para la especificación de la anchura de intervalo R se separa en K anchuras de intervalo representativas $\{Q_1, \dots, Q_K\}$, un rango de valores que se puede pre-establecer para la especificación de las probabilidades se separa en N estados de probabilidad representativos $\{P_1, \dots, P_N\}$ y se dan reglas de asignación, que asignan un Q_k ($1 \leq k \leq K$) a cada anchura de intervalo R y un P_n ($1 \leq n \leq N$) a cada probabilidad, y en el que en una segunda etapa la codificación o descodificación de los estados binarios ocurre realizando el cálculo de la nueva anchura de intervalo a obtener en el proceso de codificación o descodificación, respectivamente, utilizando una anchura de intervalo representativa Q_k ($1 \leq k \leq K$) y un estado de probabilidad representativo P_n ($1 \leq n \leq N$) mediante operaciones aritméticas distintas de la multiplicación y la división, en el que la anchura de intervalo representativa Q_k se determina mediante el intervalo de base básico de la anchura R y el estado de probabilidad representativo P_n se determina mediante la estimación de probabilidad subyacente para el símbolo que hay que codificar o a descodificar según las reglas de asignación dadas.

60 **[0032]** Por ejemplo, este tipo de programas informáticos se pueden proporcionar (a cambio de una tarifa determinada o de manera gratuita, de acceso libre o protegida por contraseña) que se puede descargar en una red de datos o de comunicación. Los programas de ordenador facilitados pueden entonces hacerse utilizables por un procedimiento en el que un programa de ordenador según la reivindicación 3 se descarga desde una red para la transmisión de datos, como por ejemplo, de Internet a unos medios de procesamiento de datos conectados a la red.

65 **[0033]** Para realizar un procedimiento para codificar y descodificar aritméticamente estados binarios se utiliza preferentemente un medio de almacenamiento legible por ordenador en el que hay un programa almacenado que permite a un ordenador, tras haber sido cargado en la memoria del ordenador, realizar un procedimiento para una

codificación y decodificación aritmética, en el que en una primera etapa un rango de valores que se puede pre-establecer para la especificación de la anchura de intervalo R se separa en K anchuras de intervalo representativas $\{Q_1, \dots, Q_K\}$, un rango de valores que se puede pre-establecer para la especificación de las probabilidades se separa en N estados de probabilidad representativos $\{P_1, \dots, P_N\}$ y se dan reglas de asignación, que asignan un Q_k ($1 \leq k \leq K$) a cada anchura de intervalo R y un P_n ($1 \leq n \leq N$) a cada probabilidad, y en el que en una segunda etapa la codificación o decodificación de los estados binarios ocurre realizando el cálculo de la nueva anchura de intervalo a obtener en el proceso de codificación o decodificación, respectivamente, utilizando una anchura de intervalo representativa Q_k ($1 \leq k \leq K$) y un estado de probabilidad representativo P_n ($1 \leq n \leq N$) mediante operaciones aritméticas distintas de la multiplicación y la división, en el que la anchura de intervalo representativa Q_k se determina mediante el intervalo de base básico de la anchura R y el estado de probabilidad representativo P_n se determina mediante la estimación de probabilidad subyacente como base para el símbolo que hay que codificar o a decodificar según las reglas de asignación dadas.

[0034] En lo que sigue, la invención se explica con más detalle con referencia a una realización ilustrada en los dibujos, en los que:

[0035]

La figura 1 muestra una ilustración de las operaciones básicas para una codificación aritmética binaria;
 La figura 2 muestra un esquema modificado de codificación aritmética asistida por tabla;
 La figura 3 muestra el principio de la decodificación aritmética asistida por tabla;
 La figura 4 muestra el principio de codificación o de decodificación, respectivamente, de datos binarios que tienen una distribución uniforme;
 La figura 5 muestra una alternativa de realización de la codificación o decodificación, respectivamente, para los datos binarios con una distribución uniforme; y
 La figura 6 muestra la inicialización de los modelos de probabilidad en función de un parámetro de cuantización SliceQP y parámetros de modelo preestablecidos m y n.

[0036] En primer lugar, sin embargo, se explica con más detalle el fondo teórico:

Estimación de probabilidad asistida por tabla

[0037] Como ya se mencionó anteriormente, el efecto de la codificación aritmética se basa en una estimación de la probabilidad de ocurrencia de los símbolos a ser codificados que debe a ser tan buena como sea posible. A fin de permitir una adaptación de estadísticas de fuentes no estacionarias, esta estimación tiene que ser actualizada en el curso del proceso de codificación. Generalmente, se utilizan para ello procedimientos habituales que operan usando contadores de frecuencia escalados de los resultados codificados [17]. Si C_{LPS} y C_{MPS} designan contadores para las frecuencias de ocurrencia de LPS y MPS, entonces utilizando estos contadores se puede realizar la estimación

$$P_{LPS} = \frac{C_{LPS}}{C_{LPS} + C_{MPS}} \quad (1)$$

y entonces se puede llevar a cabo la operación destacada en la figura 1 de la subdivisión de intervalo. A efectos prácticos, la división necesaria en la ecuación (1) es desventajosa. A menudo es conveniente y necesario, sin embargo, llevar a cabo un cambio de escala de las lecturas de contador cuando se excede un valor umbral predeterminado C_{max} del contador general $C_{Total} = C_{MPS} + C_{LPS}$. (En este contexto, es de señalar que con una representación de b-bits de L y R la probabilidad más pequeña que puede ser indicada correctamente es 2^{-b+2} , de modo que para evitar salirse de este límite inferior, si es necesario se requiere un cambio de escala de las lecturas de contador.) Con una selección adecuada de C_{max} se pueden tabular los valores recíprocos de C_{Total} , de modo que la división necesaria en la ecuación (1) puede ser sustituida por un acceso a la tabla y por una operación de multiplicación y desplazamiento. Con el fin de evitar también estas operaciones aritméticas, sin embargo, en la presente invención se utiliza un procedimiento completamente asistido por tabla para la estimación de probabilidad.

[0038] Para esto, en una fase de entrenamiento se preseleccionan los estados de probabilidad representativos $\{P_k \mid 0 \leq k < N_{max}\}$, en la que la selección de los estados es por un lado dependiente de las estadísticas de los datos que hay que codificar y por otro lado de las condiciones laterales del número máximo predeterminado N_{max} de estados. Además, se definen unas reglas de transición que indican que nuevo estado se va a utilizar para el siguiente símbolo que ha de codificarse basado en el símbolo actualmente codificado. Estas reglas de transición se proporcionan en forma de dos tablas: $\{Next_State_LPS_k \mid 0 \leq k < N_{max}\}$ y $\{Next_State_MPS_k \mid 0 \leq k < N_{max}\}$, en el que las tablas proporcionan el índice m del nuevo estado de probabilidad P_m cuando ocurre un LPS o MPS, respectivamente, para el índice n del estado de probabilidad dado actualmente. Es de señalar aquí, que para una estimación de probabilidad en el codificador o decodificador aritmético, respectivamente, como se propone en el presente documento, no se requiere la tabulación explícita de los estados de probabilidad. Más bien, los estados se asignan solamente implícitamente utilizando sus respectivos índices, como se describe en la siguiente sección. Además de

las reglas de transición se debe especificar en qué estados de probabilidad se debe intercambiar el valor de los LPS y MPS. Por lo general, sólo habrá uno de estos estados excelentes que se podrá identificar utilizando su índice p_state .

5 Separación de intervalo asistido por tabla

10 [0039] La figura 2 muestra el esquema modificado para la codificación aritmética asistida con tabla, como se propone en el presente documento. Después de la determinación de los LPS, primero de todo se asigna la anchura de intervalo R dada a un valor cuantizado Q utilizando un cartografiado tabulado Q_{tab} y una operación de desplazamiento adecuada (por q bit). Como alternativa, en algunos casos especiales la cuantización también se puede realizar sin la utilización del cartografiado tabulado Q_{tab} sólo con la ayuda de una combinación de operaciones de desplazamiento y de enmascaramiento. En general, aquí se realiza una cuantificación relativamente basta a $K = 2 \dots 8$ valores representativos. También aquí, de manera similar al caso de la estimación de probabilidad, no se realiza una determinación explícita de Q ; más bien, solamente se transfiere un índice q_index a Q . Este índice se utiliza ahora junto con el índice p_state para una caracterización del estado de probabilidad actual para la determinación de la anchura de intervalo R_{LPS} . Para ello, se utiliza ahora la entrada correspondiente de la tabla R_{tab} . Aquí, se introducen los $K \cdot N_{max}$ valores de producto $R \times P_{LPS}$, que corresponden a todos los K valores cuantizados de R y los N_{max} diferentes de los estados de probabilidad, como valores enteros con una precisión en general de $b-2$ bits. Para implementaciones prácticas se da aquí una posibilidad de sopesar entre los requisitos de almacenamiento del tamaño de la tabla y la precisión aritmética que finalmente también determina la eficiencia de la codificación. Ambas variables objetivo están determinadas por la granularidad de la representación de R y P_{LPS} .

25 [0040] En la cuarta etapa de la figura 2 se muestra cómo se realiza la actualización del estado de probabilidad p_state en función del bit de evento anteriormente codificado. Aquí, se utilizan las tablas de transición $Next_State_LPS$ y $Next_State_MPS$ que ya se mencionaron anteriormente en la sección "Estimación de probabilidad asistida por tabla". Estas operaciones corresponden al proceso de actualización que se indica en la figura 1 en la etapa 4 que no se explica con más detalle.

30 [0041] La figura 3 muestra el diagrama de flujo correspondiente de la decodificación aritmética asistida mediante tabla. Para caracterizar el intervalo parcial actual en el decodificador se utilizan la anchura de intervalo R y un valor V . Este último está presente dentro del intervalo parcial y se refina sucesivamente con cada bit leído. Como se puede ver en la figura 3, las operaciones para la estimación de probabilidad y la determinación de la anchura de intervalo R se realizan según las del codificador.

35 [0042] Codificación con distribución de probabilidad uniforme. En aplicaciones en las que, por ejemplo, unos valores con signo que hay que codificar cuya distribución de probabilidad está dispuesta simétricamente en torno a cero, para la codificación de la información de signo puede suponerse en general una distribución equitativa. Como esta información por un lado debe incrustarse en la corriente de bits aritmética, aunque por otro lado no es sensato utilizar un aparato relativamente compacto de la estimación de probabilidad y separación de intervalo asistida por tabla para el caso de una probabilidad de $p \approx 0.5$, se propone en este caso especial usar opcionalmente un procedimiento especial de codificador / decodificador que puede ser ilustrado tal como sigue.

45 [0043] En este caso especial la anchura de intervalo del nuevo intervalo parcial puede determinarse en el codificador mediante una simple operación de desplazamiento que corresponde a una bisección de la anchura del intervalo original R . En función del valor del bit que hay que codificar, la mitad superior o la mitad inferior de R , respectivamente, se selecciona entonces como un nuevo intervalo parcial (ver la figura 4). La renormalización y salida posteriores de bits se llevan a cabo como en el caso anterior de la solución asistida por tabla.

50 [0044] En el decodificador correspondiente las operaciones necesarias se reducen a determinar el bit a descodificar utilizando el valor de V en relación con la anchura de intervalo actual R mediante una simple operación de comparación. En el caso de que se establezca el bit descodificado, V debe reducirse en una cantidad de R . Como se ilustra en la figura 4, la decodificación se termina por la renormalización y la actualización de V mediante el siguiente bit que se leerá.

55 [0045] Una realización alternativa de la codificación de eventos con una distribución de probabilidad uniforme se ilustra en la figura 5. En esta implementación a modo de ejemplo no se modifica la anchura de intervalo actual R . En lugar de ello, primero se dobla V mediante una operación de desplazamiento en el codificador. En función del valor del bit que hay que codificar, entonces, de forma similar al ejemplo anterior, la mitad superior o inferior, respectivamente, de R se selecciona como un nuevo intervalo parcial (ver la figura 5). La renormalización y salida posterior de bits se realizan como en el caso anterior de la solución asistida mediante tabla con la diferencia de que la duplicación de R y L no se lleva a cabo y que las operaciones de comparación correspondientes se realizan con valores de umbral duplicados.

65 [0046] En el decodificador correspondiente de la alternativa de realización primero de todo se lee un bit y se actualiza V . La segunda etapa se realiza de la misma manera que la etapa 1 en la figura 4, es decir el bit a

descodificar se determina utilizando el valor de V en relación con la anchura de intervalo actual R mediante una simple operación de comparación, y en el caso en el que se establece el bit decodificado, V debe reducirse en la cantidad de R (ver la figura 5).

5 Acceder a e inicializar los modelos de probabilidad

10 **[0047]** Cada modelo de probabilidad, tal como se utiliza en la invención propuesta, se indica utilizando dos parámetros: 1) El índice p_state que caracteriza el estado de probabilidad de los LPS, y 2) el valor valMPS de los MPS. Cada una de estas dos variables necesita ser inicializada al comienzo de la codificación o decodificación, respectivamente, de una unidad de codificación completada (en aplicaciones de codificación de vídeo sobre una porción). Los valores de inicialización pueden por tanto derivarse de la información de control, como por ejemplo, el parámetro de cuantificación (de una porción), como se ilustra como un ejemplo en la figura 6.

15 Proceso de inicialización controlado hacia delante

20 **[0048]** Una posibilidad adicional de adaptación de las distribuciones de partida de los modelos se proporciona por el siguiente procedimiento. Con el fin de garantizar una mejor adaptación de las inicializaciones de los modelos, en el codificador se puede proporcionar una selección de valores iniciales predeterminados de los modelos. Estos modelos se pueden combinar en grupos de distribuciones de partida y se puede acceder a estos usando índices, de modo que en el codificador se lleva a cabo la selección adaptativa de un grupo de valores de partida y se transmite al decodificador en la forma de un índice como información de página. Este procedimiento se conoce como un proceso de inicialización controlado hacia delante.

25 **[0049]** Por último, se señala una vez más que hay la posibilidad de que el número de posibles índices de cuantificación y / o el número de estados de probabilidad pueda ser seleccionado como una función de la precisión predefinida de la codificación y / o como una función del espacio de almacenamiento disponible. La estimación de probabilidad de los estados puede llevarse a cabo mediante una máquina de estados finitos (FSM). Además, es posible que los estados de probabilidad se generen fuera de línea. Por último, se puede prever que la selección de los estados dependa de la estadística de los datos que hay que codificar y/o del número de estados.

30

Referencias

[0050]

- 5 [1] T. Wiegand, G. Sullivan, "Draft Text of Final Draft International Standard (FDIS) of Joint Video Specification (ITU-T Rec. H. 264 / ISO/IEC 14496-10 AVC)", JVT-G050, March 2003.
- [2] D. A. Huffman, "A method for Construction of Minimum Redundancy code", Proc. IRE, Vol. 40, pp. 1098-1101, 1952.
- 10 [3] I. H. Witten, R. M. Neal, J. G. Cleary, "Arithmetic Coding for Data Compression", Communication of the ACM, Vol. 30, No. 6, pp. 520-540, 1987.
- [4] G. G. Langdon, J. Rissanen, "A Simple General Binary Source code", IEEE Transactions on Information Theory, Vol. 28, pp. 800-803, 1982.
- 15 [5] C. E. Shannon, "A Mathematical Theory of Communication", Bell Syst. Tech. Journal, vol. 27, pp. 379-423, 623-656, 1948.
- [6] P. Elias, in "Information Theory y Coding", N. Abramson (Ed.), New York, Mc-Gra-Hill, 1963.
- [7] J. Rissanen, "Generalized Kraft Inequality y Arithmetic Coding", IBM J. Res. Develop., Vol. 20, pp. 198-203, 1976.
- 25 [8] R. C. Pasco, "Source Coding y Algorithms for Fast Data Compression", Ph. D. Dissertation, Stanford University, USA, 1976.
- [9] G. G. Langdon, "An Introduction to Arithmetic Coding", IBM J. Res. Develop., Vol. 28, pp. 135-149, 1984.
- 30 [10] A. Moffat, R. M. Neal, I. H. Witten, "Arithmetic Coding Revisited", Proc. IEEE Data Compression Conference, Snowbird (USA), pp. 202-211, 1996.
- [11] J. Rissanen, K. M. Mohiuddin, "A Multiplication-Free Multialphabet Arithmetic Arithmetic Code", IEEE Trans. on Communication, Vol. 37, pp. 93-98, 1989.
- 35 [12] W. B. Pennebaker, J. L. Mitchell, G. G. Langdon, R. B. Arps, "An Overview of the Basic Principles of the Q-Coder Adaptive Binary Arithmetic Coder", IBM J. Res. Develop., Vol. 32, pp. 717-726, 1988.
- [13] P. G. Howard, J. S. Vitter, "Practical Implementations of Arithmetic Coding", in "Image y Text Compression", J. Storer (Ed.), Norwell (USA), Kluwer, 1992.
- 40 [14] L. Huynh, A. Moffat, "A Probability-Ratio Approach to Approximate Binary Arithmetic Coding", IEEE Trans. on Information Theory, Vol. 43, pp. 1658-1662, 1997.
- 45 [15] D. Chevion, E. D. Karnin, E. Walach, "High-Efficiency, Multiplication Free Approximation of Arithmetic Coding", Proc. IEEE Data Compression Conference, Snowbird (USA), pp. 43-52, 1991.
- [16] G. Feygin, P. G. Gulak, P. Chow, "Minimizing Excess Code Length y VLSI Complexity in the Multiplication Free Approximation of Arithmetic Coding", Inform. Proc. Manag., vol. 30, pp. 805-816, 1994.
- 50 [17] D. L. Duttweiler, Ch. Chamzas, "Porbability estimation in Arithmetic y Adaptive-Huffman Entropy Coders", IEEE Trans. on Image Processing, Vol. 4, pp. 237-246, 1995.

55

REIVINDICACIONES

- 5 1. Un procedimiento para codificar aritméticamente un símbolo que hay que codificar que tiene un estado binario basado en una anchura de intervalo actual R y representando una probabilidad una estimación de probabilidad para el símbolo que hay que codificar, en el que la probabilidad está representada por un índice de probabilidad para asignar un estado de probabilidad entre una pluralidad de estados de probabilidad representativos, en el que el procedimiento comprende la siguiente etapa caracterizadora:
- 10 codificar el símbolo que hay que codificar realizando las siguientes sub-etapas:
- 15 cartografiar la anchura de intervalo actual con un índice de cuantización de entre una pluralidad de índices de cuantización representativos;
- 20 realizar la subdivisión de intervalo mediante el acceso a una tabla de división de intervalo utilizando el índice de cuantización y el índice de probabilidad para obtener un valor de la anchura de intervalo parcial;
- actualizar la anchura de intervalo actual utilizando el valor de la anchura de intervalo parcial para obtener una nueva y actualizada anchura de intervalo; y
- 25 renormalizar la nueva y actualizada anchura de intervalo, dando como salida un bit por operación de escalonamiento.
- 30 2. Un procedimiento para descodificar aritméticamente un símbolo codificado que tiene un estado binario basado en una anchura de intervalo actual R y representando una probabilidad una estimación de probabilidad para el símbolo codificado, en el que la probabilidad está representada por un índice de probabilidad de un estado de probabilidad entre una pluralidad de estados de probabilidad representativos, en el que el procedimiento comprende la siguiente etapa caracterizadora:
- 35 descodificar el símbolo codificado realizando las siguientes sub-etapas:
- 40 cartografiar la anchura de intervalo actual con un índice de cuantización de entre una pluralidad de índices de cuantización representativos;
- 45 realizar la división de intervalo mediante el acceso a una tabla de división de intervalo utilizando el índice de cuantización y el índice de probabilidad para obtener un valor de la anchura de intervalo parcial;
- actualizar la anchura de intervalo actual utilizando el valor de la anchura de intervalo parcial y un valor dentro de un intervalo parcial para obtener una nueva y actualizada anchura de intervalo; y
- 50 renormalizar la nueva anchura de intervalo actualizada y, con cada bit leído, refinar gradualmente el valor dentro del intervalo parcial.
3. Un programa de ordenador que permite a un ordenador tras haber sido cargado en la memoria del ordenador realizar un procedimiento según la reivindicación 1 o la 2.
4. Un medio de almacenamiento legible por ordenador en el que hay un programa almacenado que permite a un ordenador tras haber sido cargado en la memoria del ordenador realizar un procedimiento según la reivindicación 1 o la 2.

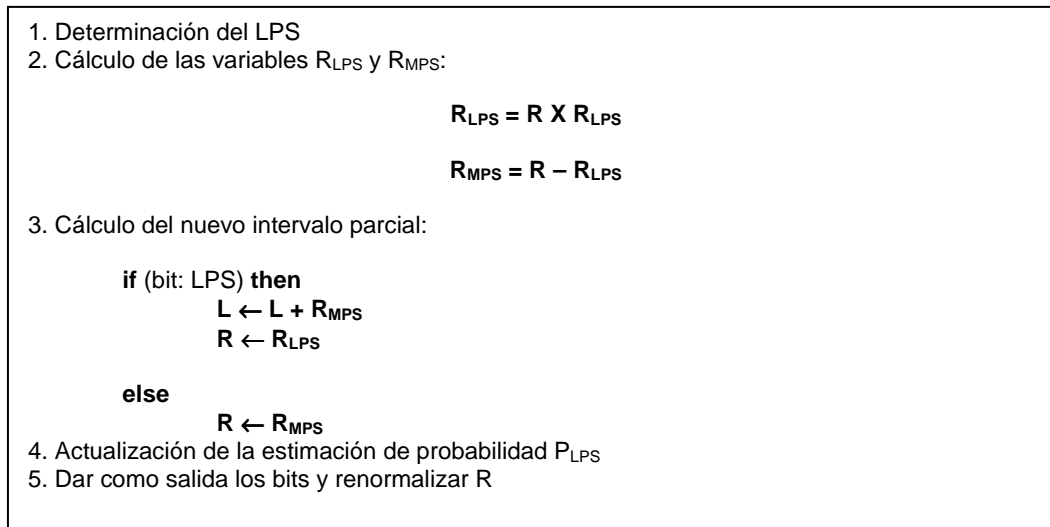


Fig. 1

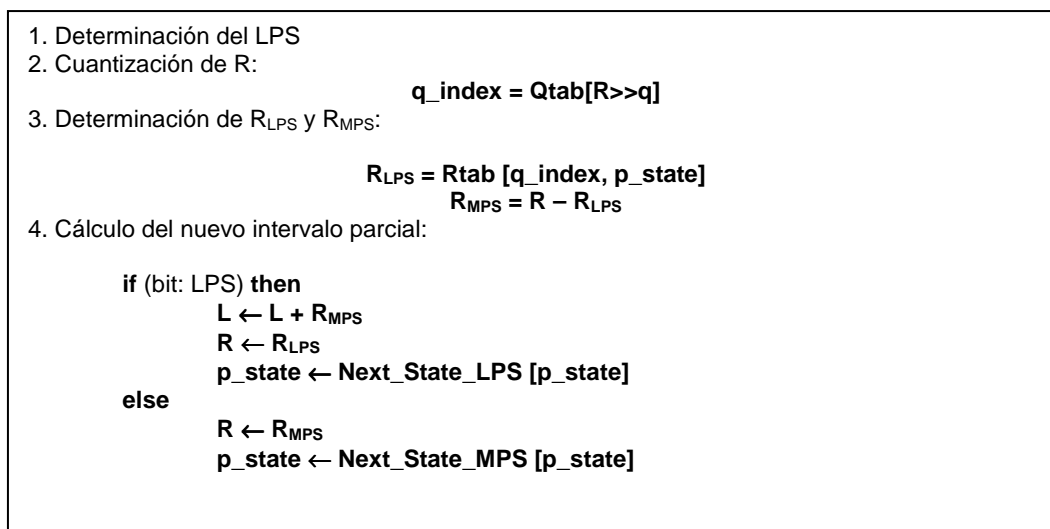
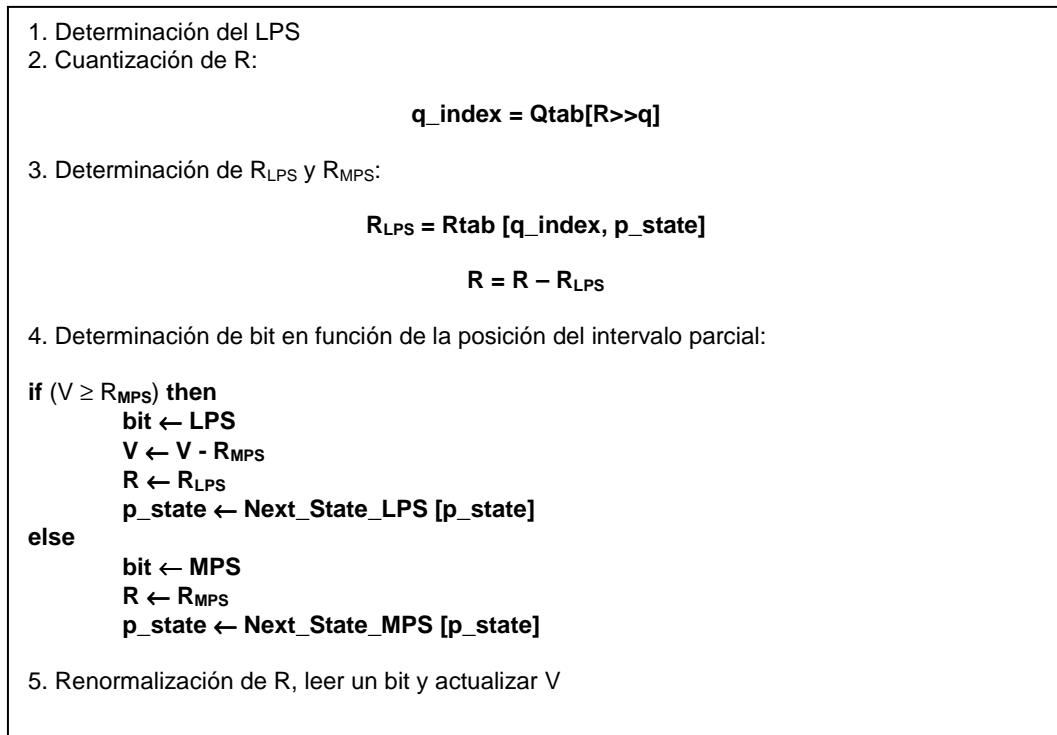


Fig. 2

Fig. 3

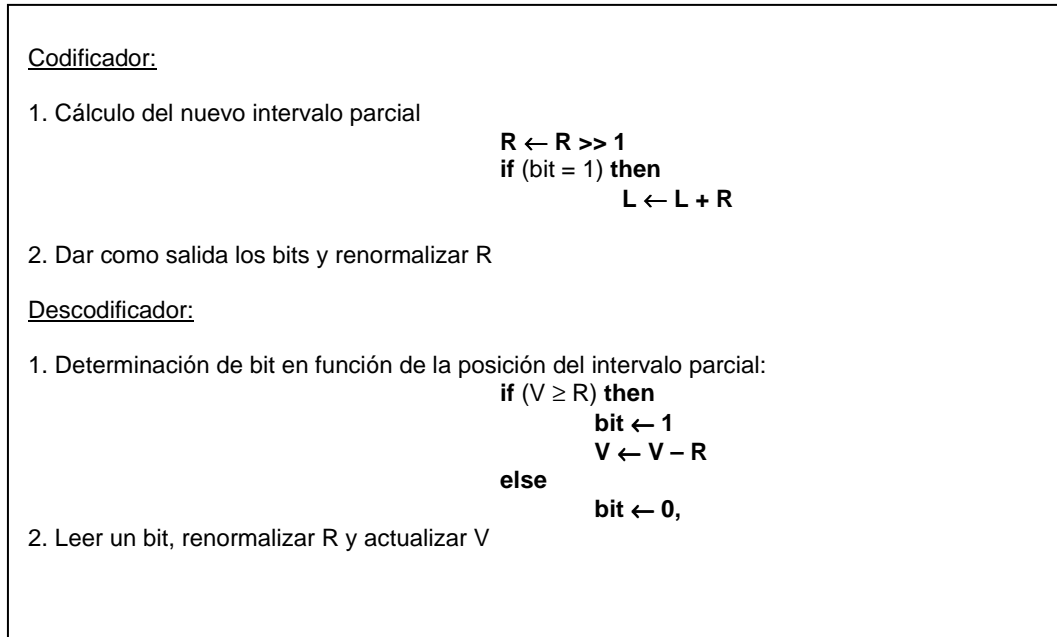


Fig. 4

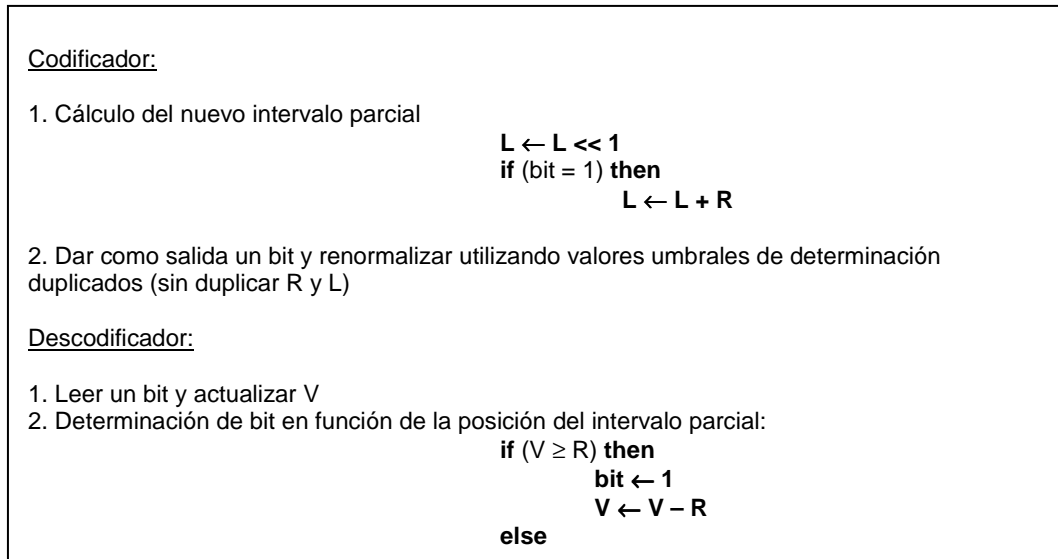


Fig. 5

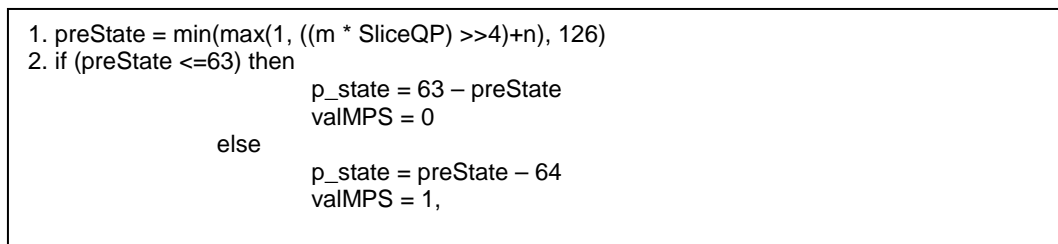


Fig. 6