



19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

11 Número de publicación: **2 367 747**

51 Int. Cl.:
H03M 7/40 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **07425116 .6**

96 Fecha de presentación : **01.03.2007**

97 Número de publicación de la solicitud: **1965497**

97 Fecha de publicación de la solicitud: **03.09.2008**

54 Título: **Método de codificación aritmética distribuida.**

45 Fecha de publicación de la mención BOPI:
08.11.2011

45 Fecha de la publicación del folleto de la patente:
08.11.2011

73 Titular/es: **SISVEL TECHNOLOGY S.R.L.**
Via Castagnole 59
10060 None, TO, IT

72 Inventor/es: **Grangetto, Marco;**
Magli, Enrico y
Olmo, Gabriella

74 Agente: **Curell Aguilá, Marcelino**

ES 2 367 747 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

Método de codificación aritmética distribuida.

5 La presente invención se refiere a un método de codificación distribuida de fuentes.

Más específicamente, la invención se refiere a un método para realizar la codificación y la decodificación de datos de comunicación que provienen de fuentes estadísticamente dependientes.

10 El documento US 2002/176494 da a conocer un método de codificación y decodificación del tipo de la presente invención.

15 En los últimos años, los académicos que se enfrentan al procesado de análisis y de señales han prestado una atención creciente a la codificación distribuida de fuentes. La codificación distribuida de fuentes considera la situación en la que dos o más fuentes de datos estadísticamente dependientes se deben codificar mediante dos codificadores independientes a los que no se permite comunicarse entre sí, es decir, cuando las dos fuentes no están disponibles de manera conjunta (es decir, no están dentro del mismo dispositivo) y, por lo tanto, la comunicación entre ellas resultaría imposible o muy cara, como en el caso de una red de sensores inalámbricos. En este caso, cada codificador tiene disponibles solamente los datos provenientes de una de las dos fuentes. Resulta también atractiva la posibilidad de aplicar principios de la codificación distribuida de fuentes a situaciones en las que una única fuente está subdividida artificialmente en sub-fuentes correlacionadas que se codifican por separado, por ejemplo, tramas diferentes de una secuencia de vídeo. En este caso, la ventaja es que ya no se requiere la decorrelación entre las sub-fuentes, simplificando de este modo el codificador (aunque a costa del decodificador).

25 Los dos codificadores pueden realizar una compresión sin pérdidas de las dos fuentes, cada uno de ellos, usando una velocidad de bits mayor que la entropía de cualquiera de las fuentes; no obstante, este tipo de compresión independiente constituye una pérdida de eficacia con respecto al uso de un codificador que comprime conjuntamente las dos fuentes, puesto que, en este caso, se podría usar una velocidad de bits igual a la entropía conjunta de las fuentes, que es menor que la entropía de cada fuente.

30 La teoría de la codificación distribuida de fuentes ha demostrado que, bajo ciertas suposiciones, se puede lograr el mismo resultado que la codificación conjunta usando dos codificadores independientes, siempre que las dos fuentes se decodifiquen por medio de un decodificador conjunto. Es posible llegar a esto realizando una codificación convencional de una de las dos fuentes (denominada "información lateral") en su valor de entropía, y una codificación "ambigua" de la otra fuente a una velocidad menor que su entropía, sin ninguna información sobre la primera fuente disponible en el segundo codificador. Una fuente se codifica de manera ambigua cuando datos de dicha fuente se transmiten en forma de una secuencia de símbolos de manera que en el decodificador no se garantiza un acuse de recibo perfecto y unívoco del símbolo transmitido.

40 Los codificadores que realizan la codificación distribuida de fuentes se implementan típicamente usando códigos de canal, es decir, representando la fuente mediante el uso de su síndrome (o los bits de paridad) para un código de canal dado adecuado para una magnitud de correlación dada entre las dos fuentes. Los códigos de canal se usan debido a que, al disponer de la información lateral, permiten realizar la decodificación de una fuente codificada de manera ambigua, con una probabilidad de error mínima.

45 La teoría de codificación distribuida de fuentes comprende no solamente la compresión sin pérdidas sino también la compresión con pérdidas; se ha observado que, bajo ciertas condiciones, no se produce ninguna pérdida de rendimiento en el uso de la codificación distribuida de fuentes por contraposición a la codificación conjunta. En la práctica, la codificación distribuida de fuentes con pérdidas se implementa típicamente usando un cuantificador convencional seguido por una codificación distribuida de fuentes sin pérdidas (codificación de entropía), mientras que el decodificador consiste en un decodificador de entropía conjunta seguido por un decuantificador conjunto.

50 Las aplicaciones de los esquemas de codificación distribuida de fuentes en señales realistas están limitadas típicamente por algunos problemas. En particular, por ejemplo, los códigos de canal requieren bloques de datos muy grandes (típicamente por encima de 10.000 símbolos). Este requisito no es siempre compatible con las aplicaciones prácticas, donde las unidades básicas a codificar son del orden de entre unos cientos y unos miles de símbolos. Además, los códigos de canal requieren que los símbolos contenidos en cada bloque sigan siempre una distribución estadística estacionaria, hipótesis que no resulta adecuada para datos como imágenes y secuencias de vídeo, por ejemplo. Por otra parte, la entropía condicional de una distribución estacionaria restringe la velocidad de bits.

60 Los códigos de canal tienen una alta complejidad de codificación, típicamente del orden del cuadrado de la dimensión del bloque de datos, y, por lo tanto, esta complejidad aumenta muy rápidamente cuando se incrementa la dimensión del bloque; por otro lado, una dimensión de bloque grande es un prerrequisito para el uso de estos códigos de canal, de lo cual se deriva la alta complejidad del proceso de codificación.

65

Un objetivo de la presente invención es proponer un método de codificación y decodificación de fuentes que pueda superar los problemas mencionados anteriormente.

5 Este y otros objetivos se logran, según la invención, con un método de codificación y decodificación cuyas características principales se definen en las reivindicaciones 1 y 7.

Otras características y ventajas de la invención resultarán evidentes a partir de la siguiente descripción detallada, proporciona únicamente a título de ejemplo no limitativo, haciendo referencia a los dibujos adjuntos, en los que:

- 10 - la figura 1 es un esquema ilustrativo de un método de codificación aritmética binaria según la técnica anterior;
 - la figura 2 es un esquema relacionado con un método de codificación distribuida aritmética binaria según la invención;
 - la figura 3 es un esquema de una rama de un árbol de decodificación aritmética distribuida; y
 15 - la figura 4 es un esquema de ramas de un árbol de decodificación aritmética distribuida que tiene una información lateral codificada de manera ambigua.

La figura 1 muestra un esquema de un método de codificación aritmética o codificación AC (Codificación Aritmética) según la técnica anterior. En la codificación aritmética, el punto de partida es un conjunto de símbolos $[X_0, X_1, \dots, X_i, \dots, X_{N-1}]$ que es una secuencia de símbolos de longitud N en la que cada símbolo pertenece a un alfabeto M -ario. El proceso de codificación aritmética se basa o bien en el conocimiento o bien en la estimación de la probabilidad de cada carácter del alfabeto M -ario en el instante de tiempo i , que se indica como $p_{1,i}, p_{2,i}, \dots, p_{M,i}$. El sumario de la probabilidad asociada a los diversos caracteres del alfabeto M -ario, en cada momento, es igual a 1. Para simplificar, el conjunto de probabilidades se indicará como p_1, p_2, \dots, p_M , aunque se pretende que estas probabilidades puedan variar en cada etapa de codificación, por ejemplo, cuando se use un algoritmo de estimación basado en estadísticas de símbolos.

La codificación aritmética se basa en la partición recursiva de intervalos según una probabilidad dada, conocida como codificación de Elías, la cual codifica símbolos a través de intervalos. Los símbolos de la fuente se asocian a sub-intervalos de un intervalo de partida de manera que los mismos constituyen una partición de este último, es decir, no se solapan y su unión es igual al intervalo de partida.

En el comienzo, el intervalo "actual" es igual al intervalo de partida y se fija a $[0,1)$. Para cada símbolo de entrada X_i de la secuencia a codificar, el intervalo "actual" se particiona en un conjunto de M sub-intervalos adyacentes cuyas longitudes son proporcionales a p_1, p_2, \dots, p_M . El sub-intervalo que representa el símbolo X_i se selecciona como el siguiente intervalo "actual".

Después de que se hayan procesado la totalidad de los N símbolos, la secuencia se representa por el último intervalo "actual", indicado como intervalo final.

La representación de la secuencia M -aria puede consistir en un número cualquiera dentro de dicho intervalo final. Por ejemplo, es posible seleccionar un punto extremo del intervalo final, el punto medio del sub-intervalo o cualquier otro número perteneciente al intervalo. Para la transmisión o para la escritura en un soporte numérico, dicho número se deberá transformar en una representación del mismo (típicamente binaria), indicada como palabra de código. En particular, para usar la velocidad de bits más baja por símbolo, el número se puede seleccionar dentro del intervalo representado por la palabra de código más corta.

Conociendo o estimando las probabilidades de los símbolos, el decodificador puede realizar las mismas particiones, y por tanto, dado el número dentro del intervalo final, puede reconstruir la secuencia de particiones o selecciones que lo han generado.

El ejemplo de la figura 1 muestra un método de codificación aritmética binaria, a saber, con M igual a 2, que codifica una cadena de $N=3$ símbolos, donde los símbolos son $X_0=0, X_1=0$ y $X_2=1$. La probabilidad del carácter 0 se indica como p_0 y la probabilidad del carácter 1 se indica como p_1 . En la figura 1, el intervalo de partida $[0,1]$ se indica como 10, que se subdivide en dos sub-intervalos proporcionales respectivamente a la probabilidad p_0 y la probabilidad p_1 .

Puesto que el primer símbolo de la secuencia a codificar es $X_0=0$, el sub-intervalo correspondiente al símbolo X_0 se selecciona dentro del intervalo de partida "actual", en este caso el sub-intervalo 11 que tiene la probabilidad p_0 . Este sub-intervalo 11 se convierte en este momento en el nuevo intervalo "actual", y se subdivide en dos sub-intervalos 12 y 13 proporcionales respectivamente a p_0 y p_1 .

En este momento, puesto que el segundo símbolo de la secuencia a transmitir es $X_1=0$, se selecciona el sub-intervalo 12 del nuevo intervalo "actual" 11 correspondiente al símbolo X_1 . A continuación, el sub-intervalo 12 se subdivide en dos sub-intervalos 14 y 15 proporcionales respectivamente a p_0 y p_1 .

Puesto que ahora el tercer símbolo de la secuencia a transmitir es $X_2=1$, el sub-intervalo 15 se selecciona como intervalo final y cualquier número binario incluido en dicho intervalo final 15 representa correctamente la secuencia $X_0 X_1 X_2$.

- 5 La codificación aritmética descrita anteriormente proporciona una descripción reversible (sin pérdida de información) de la secuencia de entrada X .

10 A continuación se dará a conocer la codificación aritmética distribuida según la invención. La misma proporciona, a diferencia de la codificación aritmética simple antes descrita, una descripción ambigua y más comprimida de la secuencia X , que se puede decodificar sin pérdidas únicamente si la información lateral Y está disponible en el receptor; dicho receptor es un decodificador conjunto suave (*soft*).

15 Las palabras de código que representan X e Y se indican como C_x y C_y , y las velocidades de codificación respectivas se indican como R_x y R_y , expresadas en velocidad de bits.

La información lateral Y se puede codificar respectivamente de manera no ambigua ($R_y \geq H(Y)$) o de manera ambigua ($R_y < H(Y)$), en la cual $H(Y)$ es la entropía de Y . El decodificador puede usar la información lateral codificada en cualquiera de los dos modos.

- 20 Por lo tanto, es posible realizar dos codificaciones diferentes, presentando la primera la información lateral disponible de manera no ambigua, presentando la segunda la información lateral disponible de manera ambigua.

25 En la codificación con información lateral disponible de manera no ambigua, la falta de ambigüedad viene dada por el número seleccionado en el intervalo final; el uso de la partición de intervalos en cada etapa del proceso de codificación aritmética garantiza que cada número real en el intervalo $[0,1]$ pertenece a solamente una de entre todas las posibles secuencias de sub-divisiones y selecciones de intervalos, correspondiéndose cada una de ellas con una secuencia de información de entrada diferente.

30 La codificación aritmética distribuida se basa, en este caso, en el principio de introducir una ambigüedad solamente en la secuencia X para permitir el uso de una menor velocidad binaria que en el caso en el que hay disponible una secuencia X no ambigua. La secuencia X se puede decodificar correctamente si se proporciona la información lateral. La ambigüedad se introduce usando uno de los siguientes tres modos.

35 En el primer modo (modo 1), la codificación aritmética distribuida utiliza una estrategia de sub-división de intervalos modificados. Se usa un conjunto de intervalos cuya amplitud se incrementa y es proporcional a un conjunto modificado de probabilidades \tilde{p}_j de tal manera que $\tilde{p}_j \geq p_j$, donde el subíndice $j=1, \dots, M$ indica los caracteres diferentes del alfabeto de codificación. Como consecuencia, la suma de las probabilidades de todos los caracteres es mayor que 1. Para hacer encajar los sub-intervalos ampliados dentro del intervalo $[0,1]$, dicho sub-intervalo se debe solapar parcialmente, tal como se muestra en la figura 2. Esto deriva en un intervalo final mayor, el cual se puede describir usando una palabra de código más corta.

40 En el modo 2 (modo 2), los codificadores aritméticos distribuidos utilizan la misma partición de intervalos que la codificación aritmética convencional, es decir, $\tilde{p}_j = p_j$. Para introducir cierta ambigüedad, se elimina un cierto número de bits de la palabra de código de salida, y dichos bits eliminados no se comunican al decodificador. Este proceso de eliminación de bits permite que el codificador utilice una velocidad de bits reducida para la codificación. Se pueden usar varios patrones de eliminación, por ejemplo, se puede eliminar un número de bits con la misma separación en las cadenas de salida, o un número de bits seleccionados aleatoriamente, o un número de bits seleccionados de tal manera que la probabilidad de error de decodificación sea mínima, y así sucesivamente.

45 En el tercer modo (modo 3), la codificación aritmética distribuida utiliza una combinación del primer y el segundo modo, es decir, los intervalos se solapan parcialmente y algunos bits se eliminan de la palabra de código de salida. El diseño de la regla de solapamiento y la regla de eliminación de bits se realiza de tal manera que se logre la velocidad de bits deseada.

50 Opcionalmente, a la palabra de código de salida se le puede añadir un código de redundancia cíclica para facilitar la detección de errores del código.

55 En la codificación con información lateral disponible de manera ambigua, se logra una codificación con una velocidad de bits próxima a la entropía conjunta codificando tanto X como Y con ambigüedad en lugar de dejar toda la ambigüedad solamente en X .

60 De hecho, el proceso de codificación aritmética distribuida se puede modificar de la manera siguiente para lograr una codificación aritmética distribuida de tiempo compartido. Si se usan los codificadores del modo 1, esto se aplica a X e Y , con la condición de que el símbolo i -ésimo X_i se codifique con intervalos solapados si y solo si el símbolo i -ésimo Y_i no se codifica con intervalos solapados. Esto se puede lograr, por ejemplo, codificando con solapamiento el

65

símbolo par de X, mientras que los símbolos impares se codifican con la partición clásica sin solapamiento. En cambio, los símbolos impares de Y se codifican con intervalo de solapamiento mientras que los símbolos pares se codifican sobre intervalos separados. Alternativamente, los símbolos a los que se aplica el solapamiento se seleccionan según una regla predeterminada (por ejemplo, una secuencia aleatoria), en la que se permite tener una proporción diferente de símbolos con solapamiento en cada secuencia.

Se puede usar la misma estrategia de tiempo compartido en el caso de la eliminación de bits (modo 2) garantizando que si se elimina un bit en una cierta posición i dentro de la palabra de código de X, no se selecciona la misma posición i para su eliminación en la palabra de código que representa Y. Este planteamiento de tiempo compartido permite diseñar un decodificador secuencial conjunto capaz de reconstruir ambas secuencias correlacionadas.

Este algoritmo se puede ampliar al caso de múltiples fuentes de información lateral Y_1, \dots, Y_M , donde la velocidad de codificación disponible está distribuida de una manera arbitraria entre las fuentes. En este caso, la codificación aritmética distribuida se aplica a cada fuente y, para cada fuente, se seleccionan los símbolos donde se usa solapamiento, así como la magnitud de solapamiento, para lograr la velocidad de bits deseada para la codificación y minimizar la probabilidad de errores de decodificación.

En la figura 2 se muestra un ejemplo de un método de codificación aritmética distribuida binaria, a saber, con M igual a 2, según una forma de realización de la invención. El método codifica una cadena de $N = 3$ símbolos, en la que los símbolos son $X_0 = 0$, $X_1 = 0$, y $X_2 = 1$. La probabilidad y la probabilidad modificada del carácter 0 se indican respectivamente como p_0 y \tilde{p}_0 , y la probabilidad y la probabilidad modificada del carácter 1 se indican respectivamente como p_1 y \tilde{p}_1 .

En dicha figura, el intervalo de partida $[0,1)$ se indica como 16, y en la inicialización se considera como el intervalo "actual". Dicho intervalo $[0, 1)$ se subdivide en dos sub-intervalos parcialmente solapados 17 y 18, proporcionales respectivamente a la probabilidad \tilde{p}_0 y la probabilidad \tilde{p}_1 ; como consecuencia de dicho solapamiento, hay presente una cierta cantidad de ambigüedad en la representación binaria final. En particular, en el lado del decodificador, siempre que la secuencia binaria apunta a la región no solapada de un cierto sub-intervalo, el símbolo de información correspondiente se puede decodificar sin ambigüedad. Por el contrario, si la palabra de código apunta a la región solapada, el símbolo fuente no se puede decodificar de manera no ambigua. En este último caso, la información lateral correlacionada se debe proporcionar al decodificador.

Puesto que el primer símbolo de la secuencia a codificar, en este caso, es $X_0 = 0$, el sub-intervalo correspondiente a símbolo X_0 se selecciona dentro del intervalo de partida "actual", en este caso, el sub-intervalo 17 que tiene la probabilidad \tilde{p}_0 . Este sub-intervalo 17 se convierte en este momento en el nuevo intervalo "actual" y se subdivide en dos sub-intervalos 19 y 20 parcialmente solapados, proporcionales respectivamente a \tilde{p}_0 y \tilde{p}_1 .

En este momento, puesto que el segundo símbolo de la secuencia a transmitir es $X_1 = 0$, se selecciona el sub-intervalo 19 del nuevo intervalo "actual" 17 correspondiente al símbolo X_1 . A continuación, el sub-intervalo 19 se subdivide en dos sub-intervalos 21 y 22, parcialmente solapados, proporcionales de manera respectiva a \tilde{p}_0 y \tilde{p}_1 .

Puesto que en este momento el tercer símbolo de la secuencia a transmitir es $X_2 = 1$, el sub-intervalo 22 se selecciona como intervalo final, y cualquier número binario dentro de dicho intervalo final 22 representa correctamente la secuencia $X_0X_1X_2$.

El método proporcionado anteriormente dado a conocer en referencia a la fuente X se puede aplicar evidentemente a la fuente Y. Los símbolos de dicha fuente Y pueden pertenecer a un alfabeto M-ario que tenga una M diferente a la del alfabeto asociado a la fuente X. En este caso, para la fuente Y, un intervalo de partida $[0, 1)$ se subdividirá en M sub-intervalos en los que M es la correspondiente asociada a la fuente Y.

La velocidad media consumida para realizar la codificación aritmética distribuida, expresada en bits por símbolo de entrada, viene dada por la siguiente ecuación:

$$R = \sum_{j=1}^M p_j \log_2 \frac{1}{\tilde{p}_j} \quad (1)$$

Como ejemplo, cada probabilidad se puede amplificar por un factor $\alpha_j \geq 1$, de lo cual se deriva que $\tilde{p}_j = \alpha_j p_j$. Como consecuencia, la velocidad de codificación media R resulta ser:

$$R = \sum_{j=1}^M p_j (\log_2 \frac{1}{p_j} - \log_2 \alpha_j) = \sum_{j=1}^M p_j (r_j - \delta r_j) \quad (2)$$

donde $r_j = -\log_2 p_j$ y $\delta r_j = \log_2 \alpha_j$, y esto último representa el número de bits eliminados de la representación binaria del símbolo i -ésimo.

5 En una forma de realización de la presente invención, los factores de amplificación α_j pueden cambiar en cada etapa de codificación, es decir, $\alpha_j = \alpha_{j,i}$, según una regla adecuada que determina la velocidad instantánea media a asignar al símbolo de entrada i -ésimo.

10 En otra forma de realización de la presente invención, puede que se desee usar la constante $\alpha_j = \alpha$. En otra forma de realización, puede que se desee mantener constante la relación $\delta r_j / r_j = k$, que es igual a fijar $\alpha_j = p_j^{-k}$, con $0 \leq k \leq 1$. En este último caso, la expresión de la velocidad de codificación media R se puede simplificar adicionalmente de la manera siguiente:

$$R = (1-k) \sum_{j=1}^M p_j \log_2 \frac{1}{p_j} = (1-k)H \quad (3)$$

15 en el que H es la entropía del modelo de probabilidad, es decir

$$H = - \sum_{j=1}^M p_j \log_2 p_j \quad (4)$$

20 Resulta evidente que la velocidad de codificación se puede reducir a cualquier valor deseado por medio de una selección adecuada del parámetro k .

A continuación se dará a conocer el método de decodificación aritmética distribuida objeto de la invención.

25 La finalidad de la decodificación aritmética distribuida es decodificar el mensaje fuente X y la información lateral correlacionada Y . En el caso más general, tanto X como Y están disponibles para el decodificador como la palabra de código codificada en entropía C_X y C_Y . El decodificador está dispuesto para realizar una búsqueda secuencial a lo largo de un árbol de decodificación apropiado donde cada nodo representa un estado del decodificador aritmético secuencial. El árbol de decodificación puede ser "accionado" por símbolos, cuando cada transición se "acciona" por la salida de un símbolo fuente dado, o "accionado" por bits, cuando cada transición representa la decodificación de un número fijo de bits codificados, correspondiente a un número variable de símbolos fuente.

30 La presente invención se aplica para cualquier decodificador que use un algoritmo apropiado de búsqueda secuencial a lo largo de un árbol de decodificación aritmética con el fin de maximizar una cierta métrica, para seleccionar el par X e Y más probable. Son posibles varias métricas; un criterio válido se basa en la métrica del Máximo A Posteriori (MAP) que se indica con $P(X_0, X_1, \dots, X_{N-1} | C_X, C_Y)$.

35 Vale la pena indicar que esta métrica abarca también el caso en el que Y se codifica usando una codificación aritmética distribuida, y se puede ampliar al caso de más de una información lateral Y .

40 La métrica MAP se puede sustituir también por otro criterio, como, por ejemplo, la distancia mínima, según el contexto o la complejidad de la decodificación deseada.

45 A continuación se darán a conocer varias formas de realización del método de decodificación. En particular, se darán a conocer diferentes métodos de decodificación en el caso de información lateral codificada de manera no ambigua o ambigua.

50 En primer lugar, se considera el caso en el que la codificación se ha realizado con solapamiento de intervalos (modo 1). En este caso se usa un árbol de codificación "accionado" por símbolos.

A continuación, se presenta la consecución de una decodificación en el caso en el que la codificación se ha realizado con eliminación de bits (modo 2). En este caso, se usa un árbol de decodificación "accionado" por bits.

55 La decodificación para datos codificados en el tercer modo (modo 3), es decir, combinación del modo 1 y 2, se puede obtener de manera directa ampliando el ejemplo presentado de decodificación.

60 Debe observarse que el método de decodificación propuesto se puede usar para decodificar más de dos fuentes correlacionadas, siempre que la métrica de decodificación tenga en cuenta todas las correlaciones. Como ejemplo, si la fuente X está correlacionada con la fuente Y_1, \dots, Y_M , su métrica de decodificación resulta ser $P(X | C_X, C_{Y_1}, \dots, C_{Y_M})$.

En una primera forma de realización del método de decodificación, se supone que la información lateral está disponible en el decodificador de manera no ambigua; esto se cumple cuando la información lateral se codifica con un planeamiento convencional (por ejemplo, la codificación de entropía convencional). La fuente correlacionada X se puede codificar con casi H (XIY) bits por símbolo, por medio de la codificación aritmética distribuida propuesta con intervalos solapados apropiadamente. En esta situación particular, el decodificador debe resolver la ambigüedad sobre X, resultante del solapamiento de intervalos, usando la información lateral Y y el modelo de correlación P (XIY).

Este objetivo se puede lograr con la decodificación distribuida secuencial que maximiza la métrica MAP P (XC_X, C_Y=Y).

En la decodificación, se procesa la palabra de código recibida C_X, que es una representación, típicamente binaria, de un número situado en el intervalo final seleccionado por el codificador. Las etapas del decodificador son las siguientes:

- 1) Inicializar el intervalo actual a [0, 1).
- 2) Subdividir el intervalo según las probabilidades \tilde{p}_j .
- 3) Inicializar el árbol de codificación.
- 4) Mientras el número de los símbolos decodificados sea menor que N (y/o se cumpla la condición de finalización):

- para cada nodo en el árbol:
 - comparar, C_X con la subdivisión de intervalo actual;
 - si C_X pertenece a la porción no solapada de un cierto intervalo, decodificar el símbolo correspondiente X_i sin ambigüedad y seleccionar el intervalo correspondiente para la siguiente iteración de este nodo;
 - si C_X se sitúa en la región solapada entre dos o más intervalos, realizar una ramificación de la decodificación; almacenar K trayectos alternativos, correspondientes a los K ≤ M símbolos alternativos X_i que se pueden decodificar, actualizar su métrica MAP y seleccionar el intervalo correspondiente para la siguiente iteración;
 - opcionalmente adoptar técnicas de poda para limitar el número de trayectos más probables almacenados en memoria; por ejemplo, se pueden usar el algoritmo de Viterbi, el algoritmo de pila y "M".

- 5) Dar salida a la secuencia correspondiente al trayecto más probable en el árbol.

En la figura 3 se ilustra un árbol de ramificación de decodificación aritmética distribuida para el caso binario. En particular, en el ejemplo presentado, el decodificador no puede tomar una decisión sobre el símbolo i-ésimo X_i, indicado por el nodo 30. Como consecuencia, se siguen dos intentos de decodificación alternativos, las ramas 31 y 32. Como en este ejemplo la fuente de entrada es binaria, se considera por un lado la hipótesis de X_i=0, rama 31, y se actualiza de forma correspondiente la métrica MAP P (X₀, X₁, ..., X_i=0|C_X, C_Y). Por otro lado se considera la decisión opuesta, rama 32, y se evalúa la métrica MAP P (X₀, X₁, ..., X_i=1|C_X, C_Y).

Dado el modelo de correlación en términos de distribución de probabilidad P(XIY), la métrica MAP se puede evaluar de manera iterativa de acuerdo con la siguiente ecuación:

$$P(X_0, X_1, \dots, X_i | C_X, C_Y) = P(X_0, X_1, \dots, X_{i-1} | C_X, C_Y) P(X_i | C_X, Y) \quad (5)$$

con:

$$P(X_i | C_X, Y) = \begin{cases} I & \text{si no se ejecutan ramas} \\ P(X_i | Y) & \text{en cualquier otro caso} \end{cases} \quad (6)$$

En la mayoría de los casos prácticos, el modelo de correlación adoptado es sin memoria, o la correlación se considera únicamente símbolo a símbolo, y, por tanto, $P(X|Y) = \prod_{i=0}^{N-1} P(X_i|Y_i)$. En este caso, el último término de la métrica MAP se puede simplificar de la manera siguiente:

$$P(X_i | C_X, Y_i) = \begin{cases} I & \text{si no se ejecutan ramas} \\ P(X_i | Y_i) & \text{en cualquier otro caso} \end{cases} \quad (7)$$

El mismo planteamiento se puede ampliar a modelos de correlación con memoria. Merece la pena observar que, por razones de complejidad, la mayoría de aplicaciones adoptarán una implementación logarítmica de la métrica previa para sustituir multiplicaciones por sumas.

En una segunda forma de realización del método de decodificación, se supone que la información lateral está disponible en el decodificador de manera ambigua. Por lo tanto, se supone que C_X y C_Y son representaciones ambiguas de fuentes correspondientes, obtenidas por medio de la codificación aritmética distribuida de tiempo compartido con solapamiento de intervalos.

En el lado del decodificador, la codificación de tiempo compartido garantiza que el árbol de codificación, correspondiente al proceso de decodificación de X e Y no presenta ramas síncronas. El proceso de decodificación conjunta de la palabra de código C_X y C_Y se puede obtener a través de las siguientes etapas:

- 1) Inicializar el intervalo actual para X, I_X a $[0, 1)$ y subdividirlo según la distribución de probabilidad de X.
- 2) Inicializar el intervalo actual para Y, I_Y a $[0, 1)$ y subdividirlo según la distribución de probabilidad de Y.
- 3) Inicializar el árbol de decodificación para X.
- 4) Inicializar el árbol de decodificación para Y.
- 5) Mientras el número de símbolos decodificados sea menor que N (y/o se cumpla la condición de finalización):
 - extender las ramas de todos los nodos en el árbol no ambiguo en el instante de tiempo i;
 - para todos los nodos en el árbol que pueden generar ambigüedad en el instante de tiempo i, fijar $Z_0^i = [Z_0, Z_1, \dots, Z_i]$ con $Z_j = X_j$ o $Z_j = Y_j$ para $0 \leq j \leq i$, según la ambigüedad resida o bien en X_i o bien en Y_i respectivamente; por otra parte, fijar $C_Z = C_X$ o $C_Z = C_Y$ basándose en la ambigüedad;
 - comparar la palabra de código C_Z con las subdivisiones del intervalo actual;
 - si C_Z pertenece a la porción no solapada de un cierto intervalo, decodificar el símbolo correspondiente Z_i sin ambigüedad, y seleccionar el intervalo correspondiente para la siguiente iteración de este nodo;
 - si C_Z se sitúa en la región solapada entre por lo menos dos intervalos, realizar una ramificación de la decodificación; almacenar $K \leq M$ trayectos alternativos, correspondientes a los símbolos alternativos Z_i que se pueden decodificar, actualizar su métrica MAP $P(Z_i^i | C_X, C_Y)$ y seleccionar el intervalo correspondiente para la siguiente iteración;
 - opcionalmente adoptar técnicas de poda para limitar el número de trayectos más probables almacenados en memoria; por ejemplo, se pueden usar el algoritmo de Viterbi, y el algoritmo de pila y "M".
- 6) Dar salida a la secuencia correspondiente al trayecto más probable en el árbol.

En la figura 4 se ilustran ramas de un árbol de decodificación aritmética distribuida en el caso en el que tanto X como Y representan fuentes binarias. Merece la pena observar que la técnica de decodificación descrita se puede aplicar también para el caso en el que símbolos de la fuente X pertenecen a un alfabeto M-ario y los símbolos de la fuente Y pertenecen a otro alfabeto M-ario con M diferente a la anterior. En particular, en el ejemplo comunicado, el decodificador no puede tomar una decisión sobre el símbolo i-ésimo Y_i , indicado por el nodo 33, y sobre el símbolo (i+1)-ésimo X_{i+1} , indicado por el nodo 34. Como consecuencia, se siguen dos intentos de decodificación alternativos, respectivamente las ramas 35 y 36 para el símbolo Y_i y las ramas 37 y 38 para el símbolo X_{i+1} . Puesto que, en este caso, la fuente de entrada es binaria, por un lado se considera la hipótesis de $Y_i=0$ y $X_{i+1}=0$, ramas 35 y 37, y se actualizan de manera correspondiente las métricas MAP $P(X_0, X_1, \dots, X_{i+1}=0 | C_X, C_Y)$ y $P(X_0, Y_1, \dots, Y_i=0 | C_X, C_Y)$; por otro lado se considera la decisión opuesta, ramas 36 y 38, y se evalúan las métricas MAP $P(X_0, X_1, \dots, X_{i+1}=1 | C_X, C_Y)$ y $P(X_0, Y_1, \dots, Y_i=1 | C_X, C_Y)$.

Por lo que a X se refiere, el decodificador no puede además tomar una decisión sobre el símbolo (i+2)-ésimo X_{i+2} , indicado por los dos nodos alternativos 39 y 40 que se derivan respectivamente de las ramas 37 y 38. Para cada uno de los dos nodos posibles, 39 y 40, se siguen dos intentos de decodificación alternativos, respectivamente las ramas 41 y 42, 41' y 42', a las cuales se pueden aplicar las mismas consideraciones explicadas anteriormente en referencia a las ramas 35, 36, 37 y 38.

El método de codificación por tanto usa dos representaciones en árbol para X e Y con el fin de hallar las dos secuencias que maximizan las métricas MAP correspondientes $P(X | C_X, C_Y)$ y $P(Y | C_X, C_Y)$. El hecho de que, en un instante de tiempo dado, exista solamente un árbol que genera decisiones ambiguas permite evaluar la métrica MAP necesaria.

Debe considerarse que, en el instante de tiempo i, el árbol de decodificación para X no contiene ambigüedad. Esto significa que las métricas MAP de todos los nodos en el árbol se mantienen invariables, por lo tanto se aplica la siguiente relación:

$$P(X_0, X_1, \dots, X_i | C_X, C_Y) = P(X_0, X_1, \dots, X_{i-1} | C_X, C_Y) \quad (8)$$

Por el contrario, el árbol para Y puede generar ramas. En tal caso, la métrica MAP se puede actualizar de la manera siguiente:

$$P(Y_0, Y_1, \dots, Y_i | C_X, C_Y) = P(Y_0, Y_1, \dots, Y_{i-1} | C_X, C_Y) P(Y_i | C_X, C_Y) \quad (9)$$

5

En ausencia de ramas, $P(Y_i | C_X, C_Y) = 1$.

En el caso opuesto, se puede usar la siguiente expresión:

$$P(Y_i | C_X, C_Y) = \sum_{m=1}^M P(Y_i | X_i = m) P(X_i = m | C_X, C_Y) \quad (10)$$

10

En la igualdad anterior, se adopta un modelo de correlación sin memoria

$$P(X|Y) = \prod_{i=0}^{N-1} P(X_i | Y_i)$$

15

por simplificar, pero el planteamiento propuesto se puede ampliar al caso con memoria.

Finalmente, el término $P(X_i = m | C_X, C_Y)$ se puede evaluar en el árbol de decodificación complementario para X según se proporciona mediante la siguiente ecuación:

20

$$P(X_i = m | C_X, C_Y) = \sum_{\text{trayectos con } X_i = m} P(X_0, X_1, \dots, X_i = m | C_X, C_Y) \quad (11)$$

Si se adoptan técnicas de poda, no todos los nodos en el árbol están disponibles y se puede usar el siguiente estimador para el término $P(X_i = m | C_X, C_Y)$:

25

$$P(X_i = m | C_X, C_Y) = \frac{\sum_{\text{trayectos con } X_i = m} P(X_0, X_1, \dots, X_i = m | C_X, C_Y)}{\sum_{\text{todos los trayectos}} P(X_0, X_1, \dots, X_i | C_X, C_Y)} \quad (12)$$

Cabe indicarse que el presente decodificador se puede generalizar al caso de múltiples fuentes correlacionadas, siempre que los árboles de todas las fuentes correlacionadas se extiendan en paralelo.

30

Si se han codificado datos con eliminación de bits (modo 2), la decodificación se puede lograr con un planteamiento que es muy similar a los casos previamente descritos. Desde el punto de vista de la implementación, es mejor realizar una búsqueda secuencial a lo largo de un árbol de decodificación "accionado" por bits. De hecho, dicho árbol presenta una ramificación para cada bit que se ha eliminado durante el proceso de codificación.

35

A continuación, para simplificar, se presenta solamente el caso con codificación no ambigua para la información lateral. Evidentemente, es posible ampliar todo lo dado a conocer en el presente documento al caso en el que también la información lateral se codifica de manera ambigua, según se ha descrito anteriormente.

40

El método de decodificación es un proceso secuencial que puede decodificar un número dado de bit de la palabra de código recibida C_X , y dar salida al número máximo de símbolos fuente. El número de símbolos fuente decodificados depende de la precisión numérica proporcionada por los dígitos binarios disponibles de C_X .

45

Las siguientes etapas permiten construir un árbol de decodificación "accionado" por bits para hallar la secuencia X, que maximiza la métrica $P(X|C_X, C_Y = Y)$.

- 1) Inicializar el intervalo actual a $[0,1)$.
- 2) Subdividir el intervalo según la probabilidad \tilde{p}_j .
- 3) Inicializar el árbol de decodificación.
- 4) Mientras no se consuman todos los dígitos de la palabra de código:

50

- para todos los nodos en el árbol:

- decodificar todos los bits de la palabra de código que preceden al siguiente bit eliminado, dar salida al símbolo fuente correspondiente y usarlos para actualizar la métrica $P(X|C_X, Y)$;

55

- ejecutar una ramificación correspondiente al bit eliminado de la palabra de código; decodificar un bit de palabra de código 0, decodificar el símbolo fuente correspondiente y la métrica actualizada; hacer lo mismo para el bit de palabra de código 1;
- 5
- opcionalmente adoptar técnicas de poda para limitar el número de trayectos más probables almacenados en memoria; por ejemplo, es posible usar los algoritmos de Viterbi, y el algoritmo de pila y "M"; es posible usar también otras restricciones para podar trayectos no adecuados, tales como el número máximo de símbolos decodificados o condiciones de finalización conocidas.
- 10
- 5) Dar salida a la secuencia correspondiente al trayecto más probable en el árbol.

La métrica se puede evaluar de manera iterativa tal como en la implementación previa. Cada trayecto en el árbol se corresponde con un número variable de símbolos fuente, obtenidos decodificando una representación truncada de la palabra de código C_x . Cada métrica del nodo se puede obtener usando el modelo de probabilidad condicional, en relación con el símbolo decodificado secuencialmente y la información lateral. En otras palabras, la métrica del nodo se debe actualizar según los símbolos fuente nuevos que deben ser decodificados, dado el aumento de la precisión de las palabras de código, cuando se decodifican uno o más bits.

15

REIVINDICACIONES

1. Método de codificación aritmética distribuida de una primera fuente de datos (X) que tiene una entropía asociada (H), y por lo menos una segunda fuente de datos (Y) correlacionada estadísticamente con la primera (X), en el que cada una de dichas fuentes (X,Y) está asociada a una secuencia respectiva de símbolos ($X_0, X_1, \dots, X_{N-1}; Y_0, Y_1, Y_{N-1}$), en la que cada símbolo pertenece a un alfabeto M-ario, en donde M puede ser diferente para las dos fuentes (X,Y), con un grupo asociado de probabilidades (p_1, \dots, p_M) cuyos valores se incluyen en el intervalo de 0 a 1; estando caracterizado el método porque se codifica un número de símbolos de dicha primera fuente (X) de manera ambigua con un número predeterminado de elementos por símbolo (X_0, X_1, \dots, X_{N-1}), en particular bits por símbolo (X_0, X_1, \dots, X_{N-1}), menor que su entropía (H(x)), y porque se codifica un número correspondiente de símbolos de dicha por lo menos segunda fuente de datos (Y) de manera no ambigua con un número predeterminado de elementos por símbolo (Y_0, Y_1, \dots, Y_{N-1}), en particular bits por símbolo (Y_0, Y_1, \dots, Y_{N-1}), mayor o igual que su entropía (H(y)).
2. Método de codificación aritmética según la reivindicación 1, en el que dicha primera fuente (X) se codifica de manera ambigua realizando las siguientes etapas:
- subdividir un intervalo de partida de números reales (16) que se supone igual al intervalo de 0 a 1, en M sub-intervalos (17, 18) en los que M adopta el valor asociado a la primera fuente (X), presentando dichos sub-intervalos longitudes predeterminadas mayores o iguales que las longitudes correspondientes a valores de las probabilidades (p_1, \dots, p_M) del alfabeto M-ario asociado a la primera fuente (X), de manera que por lo menos dos de dichos sub-intervalos (17, 18) se solapan parcialmente;
 - seleccionar un primer sub-intervalo (17) de dicho intervalo de partida (16) correspondiente al primer símbolo (X_0) de dicha secuencia;
 - repetir la etapa a) usando como nuevo intervalo de partida dicho primer sub-intervalo (17);
 - seleccionar un segundo sub-intervalo (19) de dicho intervalo de partida nuevo (17) correspondiente al segundo símbolo (X_1) de dicha secuencia (X_0, X_1, \dots, X_{N-1});
 - repetir las etapas c) y d) actualizando el intervalo de partida según los siguientes símbolos de la secuencia (X_0, X_1, \dots, X_{N-1}) hasta que se agoten todos los símbolos;
 - seleccionar cualquier número dentro del último sub-intervalo (22) correspondiente al último símbolo de la secuencia (X_0, X_1, \dots, X_{N-1}) como representativo de la secuencia completa (X_0, X_1, \dots, X_{N-1}) así codificada y transformarlo en una representación predeterminada, palabra de código indicada, constituida por una pluralidad de dichos elementos, en particular una pluralidad de bits.
3. Método de codificación según la reivindicación 1, en el que la secuencia (X_0, X_1, \dots, X_{N-1}) asociada a dicha primera fuente (X) se codifica de manera ambigua realizando las siguientes etapas:
- subdividir un intervalo de partida de números reales (16), que se supone igual al intervalo de 0 a 1 en M sub-intervalos en los que M adopta el valor asociado a la primera fuente (X), presentando dichos sub-intervalos longitudes predeterminadas iguales a las longitudes correspondientes a los valores de probabilidades (p_1, \dots, p_M) del alfabeto M-ario asociado a la primera fuente (X), de manera que dichos sub-intervalos no se solapen;
 - eliminar un número predeterminado de elementos de la palabra de código correspondiente a la secuencia (X_0, X_1, \dots, X_{N-1}).
4. Método de codificación según la reivindicación 2, en el que la secuencia (X_0, X_1, \dots, X_{N-1}) asociada a dicha primera fuente (X) se codifica de manera ambigua, realizando además la etapa siguiente:
- eliminar un número predeterminado de elementos de la palabra de código correspondiente a la secuencia (X_0, X_1, \dots, X_{N-1}).
5. Método de codificación según la reivindicación 1, en el que un número de símbolos de dicha primera fuente (X) se codifica de manera ambigua realizando las siguientes etapas:
- subdividir un intervalo de partida de números reales (16) que se supone igual al intervalo de 0 a 1, en M sub-intervalos (17, 18) en los que M adopta el valor asociado a la primera fuente (X), presentando dichos sub-intervalos longitudes predeterminadas mayores o iguales que las longitudes correspondientes a valores de las probabilidades (p_1, \dots, p_M) del alfabeto M-ario asociado a la primera fuente (X), de manera que por lo menos dos de dichos sub-intervalos (17, 18) se solapan parcialmente;
 - seleccionar un primer sub-intervalo (17) de dicho intervalo de partida (16) correspondiente a un primer símbolo (X_0) de dicha secuencia (X_0, X_1, \dots, X_{N-1});
 - repetir la etapa a) usando como nuevo intervalo de partida dicho primer sub-intervalo (17);
 - seleccionar un segundo sub-intervalo (19) de dicho intervalo de partida nuevo (17) correspondiente a un segundo símbolo (X_1) de dicha secuencia (X_0, X_1, \dots, X_{N-1});
 - repetir las etapas c) y d) actualizando el intervalo de partida según un subconjunto de los siguientes símbolos de la secuencia (X_0, X_1, \dots, X_{N-1}) hasta que se agoten todos los símbolos de dicho subconjunto;
 - seleccionar cualquier número dentro del último sub-intervalo (22) correspondiente al último símbolo de la secuencia (X_0, X_1, \dots, X_{N-1}) como representativo de la secuencia completa (X_0, X_1, \dots, X_{N-1}) así codificada y

transformarlo en una representación predeterminada, palabra de código indicada, constituida por una pluralidad de dichos elementos, en particular una pluralidad de bits; y

en el que el número correspondiente de símbolos de la segunda fuente de datos (Y) se codifica de manera no ambigua realizando las siguientes etapas:

- a') subdividir un intervalo de partida de números reales (16) que se supone igual al intervalo de 0 a 1, en M sub-intervalos, en los que M adoptó el valor asociado a la segunda fuente (Y), presentando dichos sub-intervalos longitudes predeterminadas iguales que las longitudes correspondientes a los valores de las probabilidades (p_1, p_M) del alfabeto M-ario asociado a la segunda fuente (Y), de manera que dichos sub-intervalos no se solapan;
- b') seleccionar un primer sub-intervalo (17) de dicho intervalo de partida (16) correspondiente a un primer símbolo (Y_0) de dicha segunda fuente (Y), correspondiéndose dicho primer símbolo (Y_0) de dicha segunda fuente (Y) con el primer símbolo (X_0) de la primera fuente (X);
- c') repetir la etapa a') usando como nuevo intervalo de partida dicho primer sub-intervalo (17);
- d') seleccionar un segundo sub-intervalo (19) de dicho nuevo intervalo de partida (17) correspondiente a un segundo símbolo (Y_1) de dicha segunda fuente (Y), correspondiéndose dicho segundo símbolo (Y_1) de dicha segunda fuente (Y) con el segundo símbolo (X_1) de la primera fuente (X);
- e') repetir las etapas c') y d') actualizando el intervalo de partida según un subconjunto de los siguientes símbolos de la secuencia (Y_0, Y_1, \dots, Y_{N-1}) hasta que se agoten todos los símbolos de dicho subconjunto, correspondiéndose dichos símbolos de subconjunto con los símbolos del subconjunto de la primera fuente (X);
- f) seleccionar cualquier número dentro del último sub-intervalo (22) correspondiente al último símbolo de la secuencia (Y_0, Y_1, \dots, Y_{N-1}) como representativo de la secuencia completa (Y_0, Y_1, \dots, Y_{N-1}) así codificada y transformarlo en una representación predeterminada, palabra de código indicada, constituida por una pluralidad de dichos elementos, en particular una pluralidad de bits.

6. Método de codificación según la reivindicación 3, en el que la segunda fuente de datos (Y) se codifica de manera ambigua y en el que la etapa ii) se realiza sobre un grupo predeterminado de elementos de la palabra de código correspondiente a la primera fuente (X), y la etapa iii) se realiza sobre otro grupo predeterminado de elementos de la palabra de código correspondiente a por lo menos la segunda fuente de datos (Y).

7. Método de decodificación aritmética distribuida que comprende las etapas siguientes:

- a) adquirir por lo menos una primera palabra de código (C_X) y una segunda palabra de código (C_Y) asociadas respectivamente a una primera fuente de datos (X) y a por lo menos una segunda fuente de datos (Y) correlacionada estadísticamente con la primera (X), en las que cada una de dichas palabras de código (C_X, C_Y) describe una secuencia respectiva de símbolos fuente ($X_0, X_1, \dots, X_{N-1}; Y_0, Y_1, \dots, Y_{N-1}$), en la que cada símbolo pertenece a un alfabeto M-ario, en donde M puede ser diferente para las dos fuentes (X, Y), con un grupo asociado de probabilidades (p_1, \dots, p_M); estando vinculadas dichas por lo menos dos palabras de código (C_X, C_Y) con dicha secuencia de símbolos ($X_0, X_1, \dots, X_{N-1}; Y_0, Y_1, \dots, Y_{N-1}$) por una relación de correlación predeterminada; codificándose de manera ambigua los símbolos (X_0, X_1, \dots, X_{N-1}) de la primera palabra de código (C_X) y codificándose de manera no ambigua los símbolos correspondientes (Y_0, Y_1, \dots, Y_{N-1}) de la segunda palabra de código (C_Y);
- b) subdividir por lo menos un intervalo de partida de números reales asociado a una de la primera y la segunda fuentes (X o Y), representando el intervalo de partida el intervalo desde 0 a 1, en M sub-intervalos, en los que M se corresponde con el alfabeto asociado a dicha fuente (X o Y), presentando dichos sub-intervalos longitudes predeterminadas mayores o iguales que las longitudes correspondientes a los valores de las probabilidades (p_1, \dots, p_M) del alfabeto M-ario asociado a dicha fuente (X o Y), de manera que por lo menos dos de dichos sub-intervalos se solapan parcialmente;
- c) inicializar por lo menos un árbol de decodificación constituido por un grupo de nodos que se corresponden cada uno de ellos con un símbolo de por lo menos una de las por lo menos dos palabras de código (C_X, C_Y);
- d) comparar por lo menos una de las por lo menos dos palabras de código (C_X, C_Y) con los sub-intervalos;
- e1) si dichas por lo menos palabras de código (C_X, C_Y) pertenecen a la porción no solapada de un sub-intervalo, decodificar el símbolo correspondiente y seleccionar el sub-intervalo correspondiente como un intervalo de partida nuevo;
- e2) si dichas por lo menos palabras de código (C_X, C_Y) pertenecen a la porción solapada de por lo menos dos sub-intervalos, memorizar $K \leq M$ trayectos alternativos correspondientes a los K símbolos que se pueden decodificar, actualizar su relación de correlación y seleccionar el sub-intervalo correspondiente como intervalo de partida nuevo;
- f) repetir las operaciones d) y e1) ó e2) para todos los nodos del árbol;
- g) dar salida a la secuencia correspondiente al trayecto más probable en el por lo menos un árbol de decodificación.

FIG.1

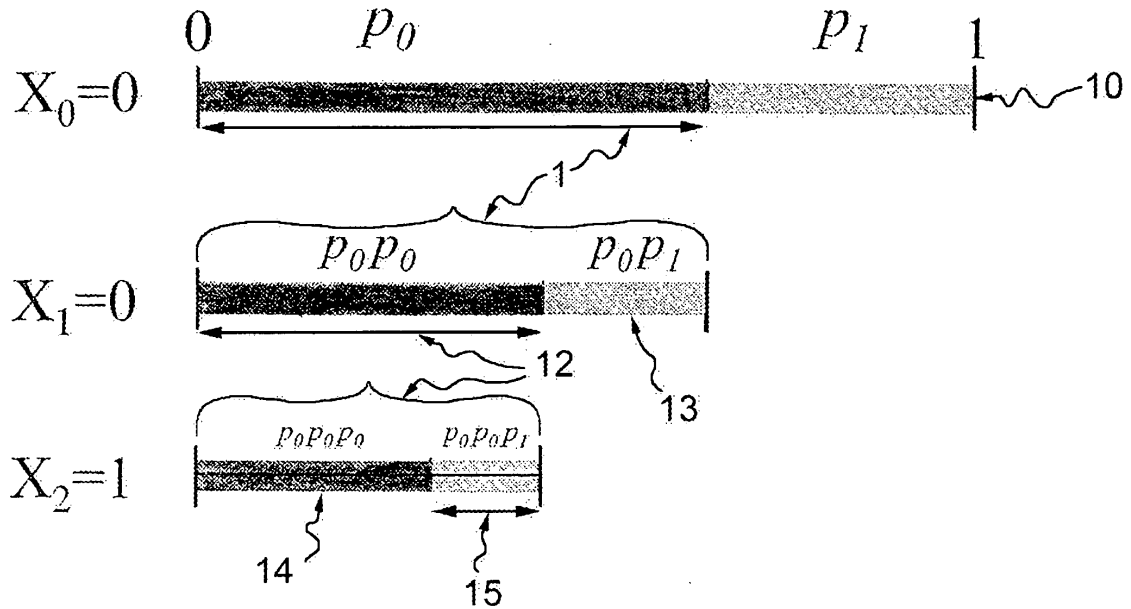


FIG.2

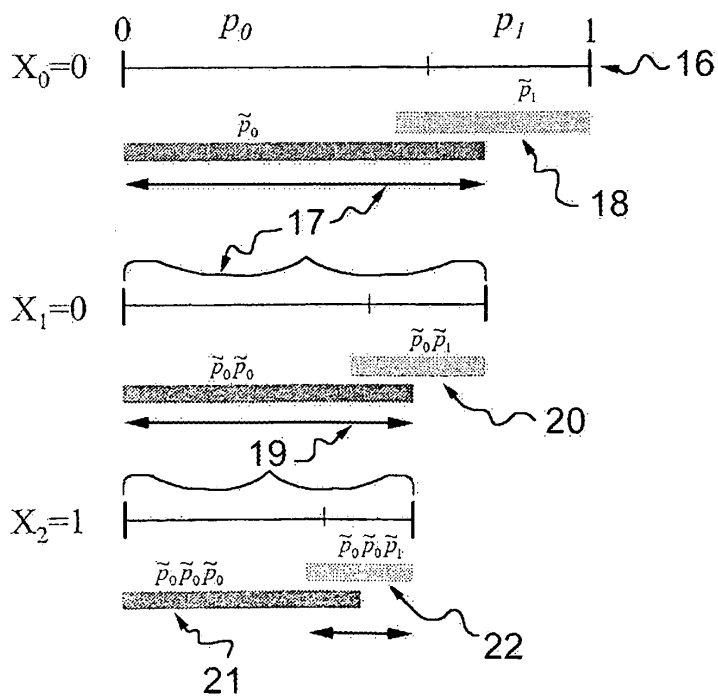


FIG.3

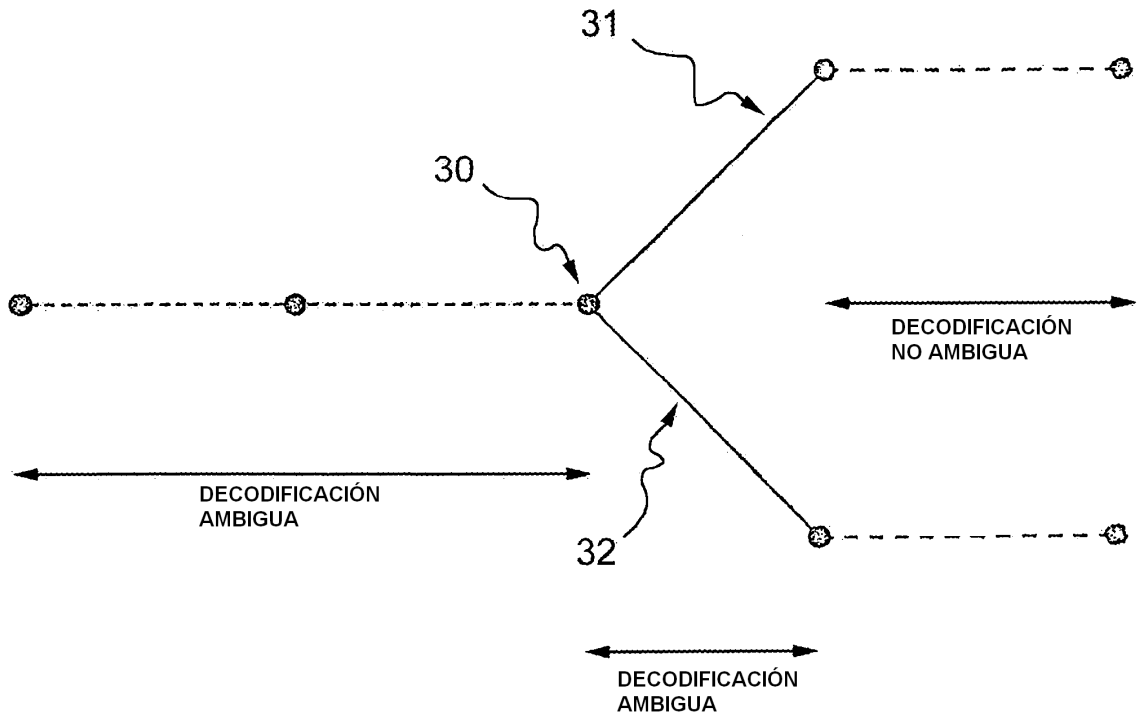


FIG.4

