



República Federativa do Brasil
Ministério do Desenvolvimento, Indústria
e do Comércio Exterior
Instituto Nacional da Propriedade Industrial.

(21) **PI0711103-7 A2**

(22) Data de Depósito: 26/04/2007
(43) Data da Publicação: 23/08/2011
(RPI 2120)



* B R P I 0 7 1 1 1 0 3 A 2 *

(51) *Int.Cl.:*
H04N 7/26 2006.01

(54) Título: **PREDIÇÃO DE MOVIMENTO ENTRE CAMADAS SIMPLIFICADAS PARA CODIFICAÇÃO DE VÍDEO ESCALONÁVEL**

(30) Prioridade Unionista: 05/05/2006 EP 06290747.2

(73) Titular(es): Thomson Licensing

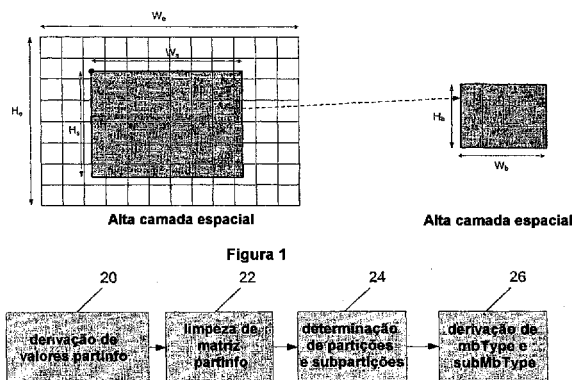
(72) Inventor(es): Christophe Chevance, Edouard Francois, Jérôme Vieron, Vincent Bottreau

(74) Procurador(es): Nellie Anne Daniel -Shores

(86) Pedido Internacional: PCT EP2007054108 de 26/04/2007

(87) Publicação Internacional: WO WO2007/128706 de 15/11/2007

(57) **Resumo:** PREDIÇÃO DE MOVIMENTO ENTRE CAMADAS SIMPLIFICADAS PARA CODIFICAÇÃO DE VÍDEO ESCALONÁVEL. Trata-se de um método para determinar a um macrobloco de alta camada que usa predição entre camadas um particionamento do macrobloco em partições. Esse compreende as seguintes etapas: - dividir o macrobloco de camada alta em blocos de camada alta não-sobrepostos de um tamanho predefinido; - determinar um pixel de camada base correspondente para um pixel, chamado pixel de referência, de cada bloco de camada alta; - identificar, para cada pixel de referência, um macrobloco de camada base ao qual o pixel de camada base correspondente pertence, uma partição de camada base à qual o dito pixel de camada base correspondente no macrobloco de camada base identificado, uma subpartição de camada base à qual o pixel de camada base correspondente pertence na dita partição de camada base identificada se existir a dita subpartição; - derivar, para cada bloco de camada alta, um único valor, chamado valor Partinfo, e - determinar um particionamento do macrobloco de camada alta em partições de macrobloco ao comparar entre esses os valores de Partinfo associados com cada bloco de alta camada.



“PREDIÇÃO DE MOVIMENTO ENTRE CAMADAS SIMPLIFICADAS PARA CODIFICAÇÃO DE VÍDEO ESCALONÁVEL”

A invenção refere-se á codificação de vídeo escalonável (SVC), mais especificamente em predição de movimento entre camadas usada em padrão SVC que é definido por Joint Video Team (JVT) de ISO/IEC MPEG & ITU-T VCEG.

Os métodos de codificação hierárquica escalonável no estado da técnica permitem a codificação das informações de maneira hierárquica para que essas possam ser decodificadas em níveis de resolução e/ou qualidade diferentes. Um fluxo de dados gerado por um dispositivo de codificação escalonável é, assim, dividido em diversas camadas, uma camada de base e uma ou mais camadas de otimização. Esses dispositivos permitem a adaptação de um único fluxo de dados a condições de transmissão variáveis (largura de banda, taxa de erro...) e também a capacidades de dispositivo de recepção (CPU, características de dispositivo de reprodução...). Um método de codificação hierárquica espacialmente escalonável (ou decodificação) codifica (ou decodifica) uma primeira parte de dados denominada de camada base se refere a imagens de baixa resolução, e a partir dessa camada base codifica (ou decodifica) ao menos outra parte de dados denominada camada de otimização que se refere a imagens de alta resolução. As informações de codificação referentes à camada de otimização são possivelmente herdadas (ou seja, derivadas) de informações de codificação referentes à camada base por um método denominado método de derivação entre camadas. As informações de codificação derivadas podem compreender possivelmente: um padrão de particionamento associado ao bloco de pixels da imagem de alta resolução (para dividir o dito bloco em diversos sub-blocos), modos de codificação associados aos ditos blocos, possivelmente vetores de movimento e um ou mais índices de referência de imagem associados com alguns blocos que permitem fazer referência à imagem usada para estimar o dito bloco. Uma imagem de referência é uma imagem da seqüência usada para prever outra imagem da seqüência. Assim, se não estiverem explicitamente codificadas no fluxo de dados, as informações de codificação (mais especificamente, tipo macrobloco e tipos sub-macrobloco) referentes à camada de otimização devem ser derivadas das informações de codificação referentes às imagens de baixa resolução. Os métodos no estado da técnica para derivar informações de codificação podem ser usados para imagens de alta resolução cujo formato não está vinculado ao formato de imagens de baixa resolução por uma transformada diádica. O método mais recente, como descrito no documento de Joint Video Team (JVT) de ISO/IEC MPEG & ITU-T VCEG JVT-R202 intitulado “Joint Scalable Video Model JSVM-5: Joint Draft 5 with proposed changes”, J.Reichel, H.Schwarz, M.Wien (indicado na seqüência JSVM5), propõe um método de predição entre camadas direta, baseado em alguns processos complexos (em termos de descrição textual).

Um objetivo da invenção é simplificar de maneira significativa esse processo de predição de movimento entre camadas e sua descrição. Mais especificamente, a invenção refere-se a um processo para determinar a divisão do dito macrobloco em partições e possivelmente o particionamento das ditas partições em subpartições. Outro objetivo da invenção é derivar o tipo de macrobloco e tipos de sub-macrobloco, para um determinado macrobloco de alta camada, utilizando predição entre camadas, de acordo com a reivindicação 1.

Outras particularidades e vantagens da invenção se tornarão claras a partir da seguinte descrição determinada a título de exemplos sem caráter limitativo e determinada em consideração às figuras em anexo que representam:

- figura 1, relações geométricas entre imagens espaciais baixas e altas,
- figura 2, um sinótico de particionamento de macrobloco, subparticionamento, processos de derivação de tipo de macrobloco e tipos de sub-macrobloco.
- figura 3, possíveis índices de partição e subpartição,
- figura 4, sinótico de processo de limpeza de blocos 4x4,
- figura 5, sinótico de limpeza de blocos 8x8,
- figura 6, sinótico de processos de particionamento de macrobloco e derivação de mbType; e
- figura 7, sinótico de processo de subparticionamento e derivação de subMbTypes.

Na seqüência, duas camadas espaciais são consideradas, uma camada baixa (denominada camada base) correspondente às imagens de baixa resolução e uma camada alta (denominada camada de otimização) correspondente às imagens de alta resolução. Como representado na figura 1, a largura e altura de imagens da camada de otimização (ou seja, imagens de alta resolução) são definidas respectivamente por W_e e H_e . As dimensões de imagens da camada base (ou seja, imagens de baixa resolução) são definidas por W_b e H_b . As imagens de baixa resolução pode ser uma versão com resolução reduzida de sub-imagens de imagens de camada de otimização, de dimensões W_s e H_s , posicionadas em coordenadas (X_{orig}, Y_{orig}) no sistema de coordenadas de imagens de camada de otimização. As imagens de baixa e alta resolução também podem ser proporcionadas por câmeras diferentes. Nesse caso, as imagens de baixa resolução não são obtidas ao reduzir a resolução de imagens de alta resolução e os parâmetros geométricos podem ser proporcionados por meios externos (por exemplo, pelas próprias câmeras). A janela correspondente à camada base é denominada uma janela de recorte. Um macrobloco da imagem de baixa resolução corresponde a um macrobloco da imagem de alta resolução quando se sobrepõe a parte de imagem de baixa resolução de resolução aumentada pela razão entre camadas em ambas as direções com a parte de imagem de alta resolução, o macrobloco correspondente cobre ao menos parcialmente o dito macrobloco da imagem de

alta resolução. Nas bordas da imagem de camada de otimização, os macroblocos podem não possuir camada base correspondente ao macrobloco, bem serem apenas parcialmente cobertos pelos macroblocos de camada base escalonados.

5 Para um determinado macrobloco de alta camada que usa predição entre camadas, as partições, possivelmente subpartições, o processo de derivação de tipo de macrobloco e tipos de sub-macrobloco está baseado no uso de uma matriz de números inteiros 4x4, denominada $partInfo[x][y]$, com $x=0$ a 3 e $y=0$ a 3.

O processo trabalha nas seguintes etapas:

- 10
- na etapa 20, a derivação de valores de matriz $partInfo$,
 - na etapa 22, possivelmente a limpeza de matriz $partInfo$,
 - na etapa 24, a determinação de partições e possivelmente subpartições; e
 - na etapa 26, a derivação de tipo de macrobloco e tipos de sub-macrobloco. Essas etapas sucessivas, representadas na figura 2, são detalhadas mais adiante.

15 **Derivação de valores de $partInfo$ (etapa 20)**

Seja (x_M, y_M) a posição superior esquerdo do dito macrobloco de alta camada na referência de alta camada (com a origem posicionada no pixel superior esquerdo da imagem de alta camada).

De acordo com a invenção, o processo compreende uma etapa para identificar a cada bloco 4x4 (x, y) , $x=0$ a 3 e $y=0$ a 3 do dito macrobloco de alta camada um macrobloco correspondente $mbBase$ na imagem de camada base, ou seja, identificar seu endereço $mbAddrBase$. Para esse propósito, seja $(X_{in}=4*x, Y_{in}=4*y)$ a posição de pixel de entrada desse processo, que irá derivar o endereço de macrobloco de camada base correspondente, o índice de partição de camada base do dito macrobloco de camada base, e o índice de subpartição de camada base da dita partição, que compreende o pixel correspondente na camada base de pixel (X_{in}, Y_{in}) . Seja (x_P, y_P) o autodenominado pixel de referência do bloco 4x4 do dito macrobloco de alta camada definido como

$$o \ x_P = X_{in} + d$$

$$o \ y_P = Y_{in} + d$$

30 onde d é um parâmetro predefinido ajustado no intervalo $[0,3]$. Em uma implementação preferida, d é ajustado para 1.

A posição correspondente (x_B, y_B) na camada base é então computada da seguinte maneira:

$$o \ x_B = ((x_P + x_M - X_{orig}) * W_b + W_s / 2) / W_s$$

35
$$o \ y_B = ((y_P + y_M - Y_{orig}) * H_b + H_s / 2) / H_s$$

O endereço $mbAddrBase$ do macrobloco de camada base $mbBase$ que compreende o pixel (x_B, y_B) é identificado como se segue:

$$o \text{ mbAddrBase} = (y_B / 16) * (W_b / 16) + (x_B / 16)$$

Se o dito macrobloco de camada base mbBase não existir (por exemplo, o pixel (xB,yB) está fora dos limites de imagem de camada base), mbAddrBase é marcado como não disponível. A predição entre camadas é proibida para o macrobloco de alta camada e o processo é interrompido.

O processo compreende uma etapa para identificar a partição (mbPartIdxBase) e subpartição (subMbPartIdxBase) de mbBase de bloco correspondente 4x4. A partição (mbPartIdxBase) e subpartição (subMbPartIdxBase) são identificadas cp,p se segue:

- mbPartIdxBase é ajustado para o índice da partição de mbBase que compreende pixel (xB,yB).

- - subMbPartIdxBase é derivado como se segue:

o Se mbBase for dividido como 8x8 (dividido em 4 blocos 8x8), subMbPartIdxBase é ajustado para o índice da subpartição de partição mbPartIdxBase de mbBase que compreende pixel (xB,yB).

o De outra forma, subMbPartIdxBase é ajustado para 0.

O processo de derivação de macrobloco base, partição e subpartição descrito acima é aplicado com a posição (Xin=4*x,Yin=4*y) como entrada, para derivar o endereço de macrobloco base mbAddrBase, o índice mbPartIdxBase da partição de camada base e o índice subMbPartIdxBase da subpartição, correspondente ao dito bloco 4x4. A Figura 3 fornece os possíveis índices de partição e subpartição, dependendo do particionamento de macrobloco, como definido na figura 6 a 9 de JSVM5. Essa refere-se, por exemplo, a partições de um macrobloco de amostras claras 16*16 e amostras de pureza de cor associadas e a partições de um sub-macrobloco de amostras claras 8*8 e amostras de pureza de cor associadas. O processo compreende uma etapa para derivar partInfo[x][y]. partInfo[x][y] é computado como se segue:

o Se o macrobloco base de endereço mbAddrBase for INTRA codificado

- partInfo[x][y] = -1 (ou seja, é marcado como derivado de macrobloco base intra), um valor negativo exceto -1 pode ser usado para identificar que o bloco é herdado de macrobloco base intra

o De outro modo

- partInfo[x][y] = 16* mbAddrBase + 4* mbPartIdxBase + subMbPartIdxBase.

Essa fórmula é definida para que um único conjunto (mbAddrBase, mbPartIdxBase, subMbPartIdxBase) gere um único valor de partInfo.

A partir de partInfo[x][y], mbAddrBase, mbPartIdxBase e subMbPartIdxBase podem ser especificamente identificados como:

$$\text{mbAddrBase} = \text{partInfo}[x][y] / 16$$

$$\text{mbPartIdxBase} = (\text{partInfo}[x][y] \% 16) / 4$$

subMbPartIdxBase = (partInfo[x][y] % 16) % 4

Limpeza de matriz de partInfo (etapa 22)

O seguinte processo é então aplicado para gerenciar as configurações em que alguns blocos 4x4 são herdados de macroblocos base intra e outros de macroblocos base inter. O processo limpa a matriz partInfo ao substituir qualquer elemento de partInfo[x][y] igual a -1 por um de seu elemento adjacente. Esse processo de limpeza se aplica primeiro a blocos 4x4 de cada bloco 8x8. Então, os blocos 8x8 são limpos, se necessário. O processo é detalhado na seqüência. Para cada bloco 8x8 (X,Y), X=0 a 1 e Y= 0 a 1, o seguinte se aplica:

10 **Limpeza de blocos 4x4**

O sinótico do processo é ilustrado na figura 4. Se o valor de partInfo de um determinado bloco for igual a -1, esse é substituído pelo valor de partInfo de um de seus blocos 4x4 adjacentes que pertence ao mesmo bloco 8x8, se esse valor não for igual a -1. Os blocos 4x4 adjacentes são varridos na seguinte ordem: 1. adjacente horizontal, 2. adjacente vertical, 3. adjacente diagonal.

Descrição de pseudocódigo desse processo

- Na etapa 40, para cada bloco 4x4 (x,y), x=0 a 1 e y=0 a 1, o seguinte se aplica.

- Se partInfo[2 * X + x][2 * Y + y] for igual a -1, o seguinte se aplica.

- Na etapa 42, se partInfo[2 * X + 1 - x][2 * Y + y] não for igual a - 1, cópia de adjacente horizontal: partInfo[2*X + x][2*Y + y] = partInfo[2 * X + 1 - x][2 * Y + y]

- De outro modo, na etapa 44, se partInfo[2 * X + x][2 * Y + 1 - y] não for igual a -1, cópia de adjacente vertical: partInfo[2 * X + x][2 * Y + 1 - y] = partInfo[2 * X + x][2 * Y + 1 - y]

- De outro modo, na etapa 46, se partInfo[2 * X + 1 - x][2 * Y + 1 - y] não for igual a -1, cópia de adjacente diagonal: partInfo[2 * X + x][2 * Y + y] = partInfo[2 * X + 1 - x][2 * Y + 1 - y]

25 **Limpeza de bloco 8x8**

O sinótico do processo é ilustrado na figura 5. Para um determinado bloco 8x8, se o valor de partInfo de seu primeiro bloco 4x4 for igual a -1, os valores de partInfo de seus 4 blocos 4x4 são substituídos por aqueles de um bloco 8x8 adjacente que pertence ao macrobloco de alta camada. Essa substituição é apenas aplicada se o primeiro bloco 4x4 do bloco 8x8 adjacente considerado possuir um valor de partInfo que não seja igual a -1. Os blocos 8x8 adjacentes são varridos na seguinte ordem: 1. adjacente horizontal, 2. adjacente vertical, 3. adjacente diagonal.

Descrição de pseudocódigo desse processo

- Na etapa 50, para cada bloco 8x8 (X,Y), X=0 a 1 e Y=0 a 1, o seguinte se aplica.

- Na etapa 52, se partInfo[2 * X][2 * Y] for igual a -1, o seguinte se aplica.

- Se partInfo[2 * (1 - X)][2 * Y] não for igual a -1, cópia de adjacente horizontal:

- Para cada bloco 4x4 (x,y), x=0 a 1 e y=0 a 1, o seguinte se aplica.

$$\text{partInfo}[2 * X + x][2 * Y + y] = \text{partInfo}[2 * (1 - X)][2 * Y]$$

- De outro modo, na etapa 54, se $\text{partInfo}[2 * X][2 * (1 - Y)]$ não for igual a -1, cópia de adjacente vertical:

- Para cada bloco 4x4 (x,y), x=0 a 1 e y=0 a 1, o seguinte se aplica.

$$\text{partInfo}[2 * X + x][2 * Y + y] = \text{partInfo}[2 * X][2 * (1 - Y)]$$

- De outro modo, na etapa 56, se $\text{partInfo}[2 * (1 - X)][2 * (1 - Y)]$ não for igual a -1,

cópia de adjacente diagonal:

- Para cada bloco 4x4 (x,y), x=0 a 1 e y=0 a 1, o seguinte se aplica.

$$\text{partInfo}[2 * X + x][2 * Y + y] = \text{partInfo}[2 * (1 - X)][2 * (1 - Y)]$$

10 Derivação de partições, subpartições, tipo de macrobloco (mbType) e tipos de sub-macrobloco (subMbType)

Esse processo deriva o macrobloco de camada alta tipo mbType e, se esse for dividido em 4 blocos 8x8, o sub-macrobloco tipo subMbType[blkIdx] de cada bloco 8x8 de índice blkIdx, a partir de matriz de partInfo e do tipo de macroblocos de camada base (também denominados, modos de codificação).

15

Dados de camada base disponíveis

Considera-se que cada macrobloco de camada base contém uma informação disponível que indica a cada bloco 8x8 qual lista de predição esse utiliza. Essa informação é armazenada como uma matriz bidimensional usedPredList[mbAddr][blkIdx], sendo que a primeira dimensão é correspondente ao endereço de macrobloco mbAddr, a segunda dimensão ao índice do bloco 8x8 dentro do macrobloco blkIdx (obtendo valores entre 0 e 3).

20

Se o bloco 8x8 de índice blkIdx de macrobloco de endereço mbAddr usar:

- apenas lista de predição L0 $\text{usedPredList}[\text{mbAddr}][\text{blkIdx}] = 1.$

- apenas lista de predição L1, $\text{usedPredList}[\text{mbAddr}][\text{blkIdx}] =$

25

2.

- ambas as listas de predição L0 e L1, $\text{usedPredList}[\text{mbAddr}][\text{blkIdx}] = 3.$

Como especificado em JSVM5, o tipo de macrobloco é definido entre os valores da Tabela 1. Similarmente, como especificado em JSVM5, o tipo de sub-macrobloco é definido entre os valores da Tabela 1.

Tipo de macrobloco	
B_L0_16x16	B_L1_Bi_16x8
B_L1_16x16	B_L1_Bi_8x16
B_Bi_16x16	B_Bi_L0_16x8

Tipo de sub-macrobloco	
B_L0_8x8	B_Bi_4x8
B_L1_8x8	B_L0_4x4
B_Bi_8x8	B_L1_4x4

B_L0_L0_16x8	B_Bi_L0_8x16	B_L0_8x4	B-Bi_4x4
B_L0_L0_8x16	B_Bi_L1_16x8	B_L1_8x4	P-L0_8x8
B_L1_L1_16x8	B_Bi_L1_8x16	B_Bi_8x4	P_L0_8x4
B_L1_L1_8x16	B_Bi_Bi_16x8	B_L0_4x8	P_L0_4x8
B_L0_L1_16x8	B_Bi_Bi_8x16	B_L1_4x8	P_L0_4x4
B_L0_L1_8x16	B_8x8		
B_L1_L0_16x8	P_L0_16x6		
B_L1_L0_8x16	P_L0_L0_16x8		
B_L0_Bi_16x8	P_L0_L0_8x16		
B_L0_Bi_8x16	P_8x8		

Tabela 1: possíveis tipos de macrobloco e sub-macrobloco.

Descrição de processo

O processo de derivação de tipo de macrobloco mbType e, se aplicável, tipo de sub-macrobloco subMbType[blkIdx] de cada bloco 8x8 de índice blkIdx, funciona como se segue.

Determinação de partição e derivação de mbType

O sinótico do processo é ilustrado na figura 6.

- Na etapa 60, se todos os valores de partInfo forem iguais a -1, mbType é ajustado para modo I_BL.

- De outro modo, o particionamento é determinado dependendo da análise de valores de partInfo.

o Se todos os 16 valores de partInfo forem iguais,

▪ Na etapa 62, o particionamento partSize é ajustado para 16x16

▪ Na etapa 64, derivação de informações de lista de predição

• partPredMode0 = usePredList[partInfo[0][0]/16][(((partInfo[0][0]%16)/4)].

o De outro modo, se 8 valores de partInfo superiores forem iguais aos 8 valores de partInfo inferiores são iguais,

▪ na etapa 66, o particionamento partSize é ajustado para 16x8

▪ Na etapa 68, derivação de informações de lista de predição de partição acima de

16x8:

• partPredMode0 = usePredList[partInfo[0][0]/16][(((partInfo[0][0]%16)/4)].

▪ Na etapa 68, derivação de informações de lista de predição de partição abaixo de 16x8:

- $\text{partPredModel} = \text{usePredList}[\text{partInfo}[3][3]/16][(((\text{partInfo}[3][3]\%16)/4)]$.

o De outro modo, se os 8 valores de partInfo da esquerda forem iguais e os 8 valores de partInfo da direita forem iguais,

- Na etapa 70, o particionamento partSize é ajustado para 8x16

▪ Na etapa 72, derivação de informações de lista de predição de partição 8x16 esquerda:

- $\text{partPredMode0} = \text{usePredlist}[\text{partInfo}[0][0]/16][(((\text{partInfo}[0][0]\%16)/4)]$.

10 ▪ Na etapa 72, derivação de informações de lista de predição de partição 8x16 direita:

- $\text{partPredModel} = \text{usePredlist}[\text{partInfo}[3][3]/16][(((\text{partInfo}[3][3]\%16)/4)]$.

o De outro modo,

- Na etapa 74, o particionamento partSize é ajustado para 8x8

15 Por fim, o mbType é derivado, na etapa 76, com base no particionamento de partSize, as informações de lista de predição derivadas partPredMode0 e partPredModel e o tipo de porção slice_type que compreende o macrobloco de alta camada sob consideração, utilizando a Tabela 2.

slice_type	partSize	partPredMode0	partPredMode1	mbType	slice_type	partSize	partPredMode0	partPredMode1	mbType
B	16x16	< 2	na	B_L0_16x16	B	16x8	2	3	B_L1_Bi_16x8
B	16x16	2	na	B_L1_16x16	B	8x16	2	3	B_L1_Bi_8x16
B	16x8	3	Na	B_Bi_16x16	B	16x8	3	< 2	B_Bi_L0_16x8
B	16x8	< 2	< 2	B_L0_L0_16x8	B	8x16	3	< 2	B_Bi_L0_8x16
B	8x16	< 2	< 2	B_L0_L0_8x16	B	16x8	3	2	B_Bi_L1_16x8
B	16x8	2	2	B_L1_L1_16x8	B	8x16	3	2	B_Bi_L1_8x16
B	8x16	2	2	B_L1_L1_8x16	B	16x8	3	3	B_Bi_Bi_16x8
B	16x8	< 2	2	B_L0_L1_16x8	B	8x16	3	3	B_Bi_Bi_8x16

B	8x16	< 2	2	B_L0_L1_8x16	B	8x8	na	na	B_8x8
B	16x8	2	< 2	B_L1_L0_16x8	P	16x16	na	na	P_L0_16x16
B	8x16	2	< 2	B_L1_L0_8x16	P	16x8	na	na	P_L0_L0_16x8
B	16x8	< 2	3	B_L0_Bi_16x8	P	8x16	na	na	P_L0_L0_8x16
B	8x16	< 2	3	B_L0_Bi_8x16	P	8x8	na	na	P_8x8

Tabela 2: derivação de macrobloco tipo mbType ("na" significa não disponível).

Descrição de pseudocódigo desse processo

- Se $partInfo[0][0]$ for igual a -1, mbType é ajustado igual a I_BL.

5 - De outro modo, o seguinte se aplica.

- Seja $partSize$, $partPredMode0$, e $partPredModel$ variáveis temporárias que são derivadas como se segue.

- Se $partInfo[x, y]$, com $x, y = 0..3$, for igual a $partInfo[0, 0]$, o seguinte se aplica

$partSize = 16x16$

10 $partPredMode0 = usedPredictionLists [partInfo[0, 0]/ 16][(partInfo[0, 0]\%16)/4]$

- De outro modo, se $partInfo[x, y]$, com $x = 0..3$ e $y = 0..1$, for igual a $partInfo[0, 0]$ e $partInfo[x, y]$, com $x = 0..3$ e $y = 2..3$, for igual a $partInfo[0, 2]$, o seguinte se aplica.

$partSize = 16x8$

$partPredMode0 = usedPredictionLists$

15 $[partInfo[0, 0]/ 16][(partInfo[0, 0]\%16)/4]$

$partPredMode1 =$

$2 * predFlagL1Base[partInfo[3, 3]/ 16][(partInfo[3, 3] \% 16) / 4]$

- De outro modo, se $partInfo[x, y]$, com $x = 0..1$ e $y = 0..3$, for igual a $partInfo[0, 0]$ e $partInfo[x, y]$, com $x = 2..3$ e $y = 0..3$, for igual a $partInfo[2, 0]$, o seguinte se aplica

20 $partSize = 8x16$

$partPredMode0 = usedPredictionLists$

$[partInfo[0, 0]/ 16][(partInfo[0, 0]\%16)/4]$

$partPredModel = usedPredictionLists [partInfo[3, 3]/ 16][(partInfo[3, 3]\% 16) /4]$

- De outro modo, $partSize$ é ajustado igual a 8x8.

25 - Dependendo do slice_type, $partSize$, $partPredMode0$, e $partPredModel$, mbType é derivado como especificado pela Tabela 2.

Determinação de subpartições e derivação de subMbTypes

Se mbType for igual a P_8x8 ou B_8x8, o tipo de sub-macrobloco de cada bloco 8x8 é determinado, como se segue.

O sinótico do processo é ilustrado na figura 7.

Para cada bloco 8x8 (x0,y0) com x0,y0=0,...,1, o subparticionamento é determinado dependendo da análise de valores de partInfo.

- Na etapa 80, derivação de informações de lista de predição:

5 o partPredMode = usedPredictionLists [partInfo[x0, y0] / 16][(partInfo[x0, y0] % 16) / 4].

- Na etapa 82, se os 4 valores de partInfo de bloco 8x8 considerado forem iguais, o subparticionamento subPartSize é ajustado para 8x8

10 • De outro modo, na etapa 84, se 2 valores de partInfo de bloco 8x8 considerado forem iguais e 2 valores de partInfo inferiores de bloco 8x8 considerado forem iguais, o subparticionamento subPartSize é ajustado para 8x4

- De outro modo, na etapa 86, se 2 valores de partInfo da esquerda de bloco 8x8 considerado forem iguais e 2 valores de partInfo da direita de bloco 8x8 considerado forem iguais, o subparticionamento subPartSize é ajustado para 4x8

15 • De outro modo, na etapa 88, o subparticionamento subPartSize é ajustado para 4x4

20 Por fim, o subMbType do bloco 8x8 considerado é derivado, na etapa 90, baseado no subparticionamento subPartSize, as informações de lista de predição derivadas partPredMode e o tipo de porção slice_type que compreende o macrobloco de alta camada sob consideração, utilizando a Tabela 3.

Slice_type	subPartSize	partPredMode	subMbType	Slice_type	subPartSize	partPredMode	subMbType
B	8X8	< 2	B_LO_8x8	B	4X8	3	B_Bi_4x8
B	8x8	2	B_L1_8x8	B	4x4	< 2	B_LO_4x4
B	8x8	3	B_Bi_8x8	B	4x4	2	B_L1_4x4
B	8x4	< 2	B_LO_8x4	B	4x4	3	B_Bi_4x4
B	8x4	2	B_L1_8x4	P	8x8	na	P_LO_8x8
B	8x4	3	B_Bi_8x4	P	8x4	na	P_LO_8x4
B	4x8	< 2	B_LO_4x8	P	4x8	na	P_LO_4x8
B	4x8	2	B_L1_4x8	P	4x4	na	P_LO_4x4

Tabela 3: derivação de sub-macrobloco tipo subMbType (“na” significa não disponível).

Descrição de pseudocódigo desse processo

Quando mbType for igual a P_8x8 ou B_8x8, para mbPartIdx = 0..3, derivação de subMbType para cada partição é obtida como se segue.

- Seja x_0 , y_0 , e partPredMode variáveis temporárias que são derivadas por $x_0 = 2 * (mbPartIdx \% 2)$ $y_0 = 2 * (mbPartIdx / 2)$ partPredMode = usedPredictionLists

$[partInfo[x_0, y_0] / 16][(partInfo[x_0, y_0] \% 16) / 4]$

Seja subPartSize uma variável temporária que é derivada como se segue.

10 - Se partInfo[$x_0 + x$, $y_0 + y$], com $x, y = 0..1$, for igual a partInfo[x_0 , y_0], subPartSize é ajustada igual a 8x8.

- De outro modo, se partInfo [x_0 , y_0] for igual a partInfo[$x_0 + 1$, y_0] e partInfo[x_0 , $y_0 + 1$] for igual a partInfo[$x_0 + 1$, $y_0 + 1$], subPartSize é ajustado igual a 8x4.

15 - De outro modo, se partInfo [x_0 , y_0] for igual a partInfo [x_0 , $y_0 + 1$] e partInfo [$x_0 + 1$, y_0] for igual a partInfo [$x_0 + 1$, $y_0 + 1$], subPartSize é ajustado igual a 4x8.

- De outro modo, subPartSize é igual a 4x4.

Dependendo do slice_type, subPartSize, partPredMode, subMbType[mbPartIdx] é derivado como especificado pela Tabela 3.

REIVINDICAÇÕES

1. Método, como parte de um método de codificação ou decodificação de vídeo escalonável, para determinar para um macrobloco de alta camada que usa predição entre camadas um particionamento do dito macrobloco em partições, **CARACTERIZADO** pelo fato de que esse compreende as seguintes etapas:

- dividir o dito macrobloco de camada alta em blocos de camada alta não-sobrepostos de um tamanho predefinido;

- determinar um pixel de camada base correspondente para um pixel, chamado pixel de referência, de cada bloco de camada alta;

- identificar, para cada pixel de referência, um macrobloco de camada base ao qual o dito pixel de camada correspondente pertence, uma partição de camada base à qual o dito pixel de camada base correspondente no dito macrobloco de camada base identificado, uma subpartição de camada base à qual o dito pixel de camada base correspondente pertence na dita partição de camada base identificada se existir a dita subpartição;

- derivar, para cada dito bloco de camada alta, um único valor, chamado valor PartInfo, como se segue:

- ajustar o valor PartInfo a um valor negativo predefinido se o dito macrobloco de camada base identificado for INTRA codificado; e

- computar o dito valor PartInfo sobre a base do macrobloco de camada base identificado, partição de camada base identificada e subpartição de camada base identificada se a dita subpartição existir, de outro modo; e

- determinar um particionamento do dito macrobloco de camada alta em partições de macrobloco ao comparar entre esses os ditos valores de PartInfo associados com cada dito bloco de alta camada.

2. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o dito macrobloco de camada base é identificado por seu endereço mbAddrBase na imagem, sendo que a dita partição de camada base é identificada no dito macrobloco de camada base por um índice mbPartIdxBase, e em que a dita subpartição de camada base é identificada na dita partição de camada base por um índice subMbPartIdxBase.

3. Método, de acordo com a reivindicação 2, **CARACTERIZADO** pelo fato de que o dito valor PartInfo é computado de acordo com a seguinte equação: $16 * mbAddrBase + 4 * mbPartIdxBase + subMbPartIdxBase$.

4. Método, de acordo com qualquer uma das reivindicações 1 a 3, **CARACTERIZADO** pelo fato de que método compreende adicionalmente após a etapa para derivar o valor PartInfo, uma etapa para ajustar cada valor PartInfo associado com um determinado bloco de alta camada e igual ao dito valor negativo predefinido a um valor

PartInfo não igual ao dito valor negativo predefinido e associado com um dos blocos de alta camada adjacentes do dito bloco de alta camada determinado, se tal valor PartInfo existir.

5. Método, de acordo com qualquer uma das reivindicações 1 a 4, **CARACTERIZADO** pelo fato de que o dito macrobloco de alta camada possui um tamanho de 16 por 16 pixels e com isso os ditos blocos de alta camada possuem um tamanho de 4 por 4 pixels.

6. Método, de acordo com a reivindicação 5, **CARACTERIZADO** pelo fato de que:

- se todos os valores de PartInfo associados com cada bloco de alta camada forem iguais a um mesmo valor, então o dito macrobloco de alta camada é dividido em uma partição de 16 pixels de largura e 16 pixels de altura;

- se os valores de PartInfo associados com os 8 blocos de alta camada da parte superior do dito macrobloco de alta camada forem iguais a um mesmo primeiro valor e se os valores de PartInfo associados com os 8 blocos de alta camada da parte inferior do dito macrobloco de alta camada forem iguais a um mesmo segundo valor diferente do dito primeiro valor, então o dito macrobloco de alta camada é dividido em duas partições de 16 pixels de largura e 8 pixels de altura; e

- se os valores de PartInfo associados com os 8 blocos de alta camada da parte esquerda do dito macrobloco de alta camada forem iguais a um mesmo terceiro valor e se os valores de PartInfo associados com os 8 blocos de alta camada da parte direita do dito macrobloco de alta camada forem iguais a um mesmo quarto valor diferente do dito terceiro valor, então o dito macrobloco de alta camada é dividido em duas partições de 8 pixels de largura e 16 pixels de altura; e

- de outro modo, o dito macrobloco de alta camada é dividido em quatro partições de 8 pixels de largura e 8 pixels de altura.

7. Método, de acordo com a reivindicação 6, **CARACTERIZADO** pelo fato de que o dito macrobloco de alta camada é dividido em 4 partições de 8 pixels de largura e 8 pixels de altura, sendo que o método compreende ainda uma etapa para determinar, para cada partição do dito macrobloco de alta camada, um particionamento em subpartições ao comparar entre esses os ditos valores de PartInfo associados com cada dito bloco de alta camada que pertence à dita partição.

8. Método, de acordo com qualquer uma das reivindicações 1 a 7, **CARACTERIZADO** pelo fato de que o dito pixel de referência é o pixel superior esquerdo do dito bloco de alta camada.

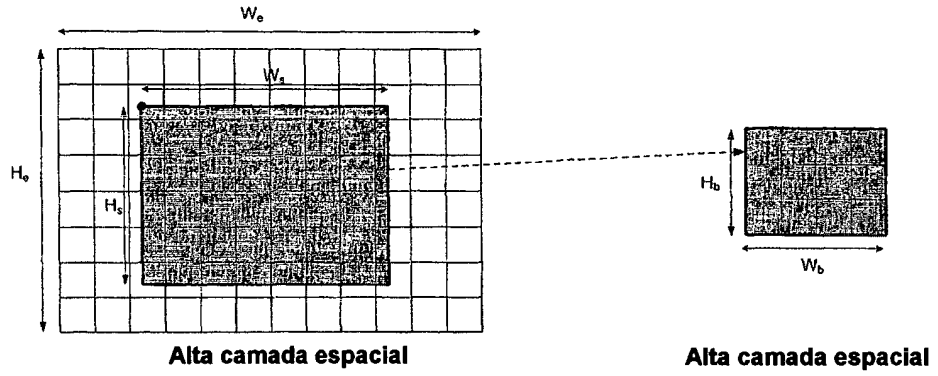


Figura 1

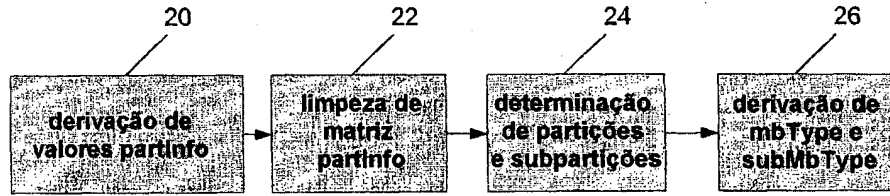


Figura 2

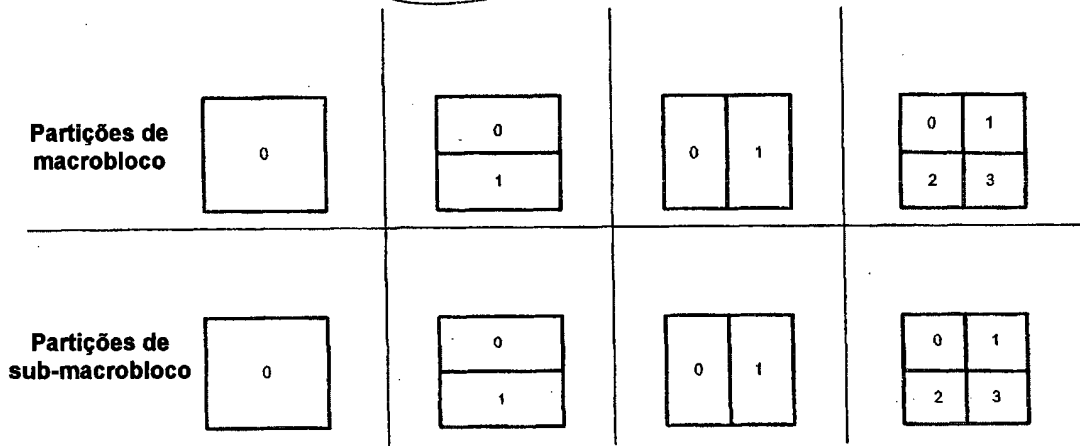


Figura 3

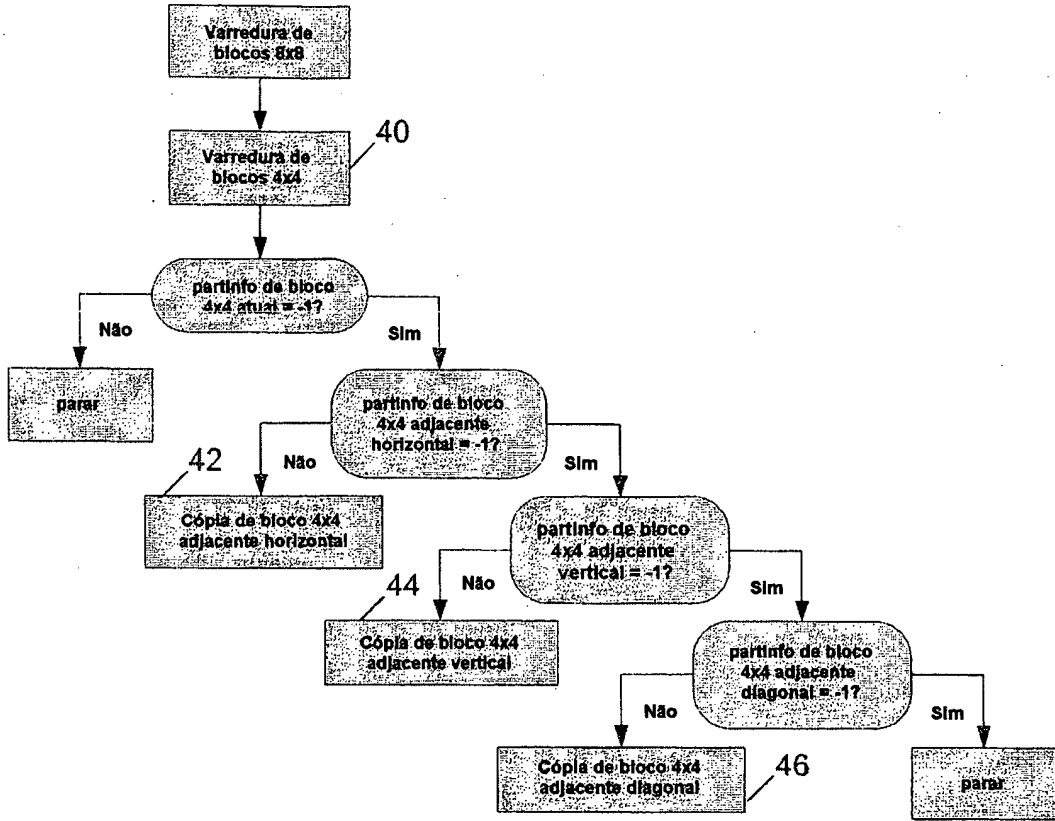


Figura 4

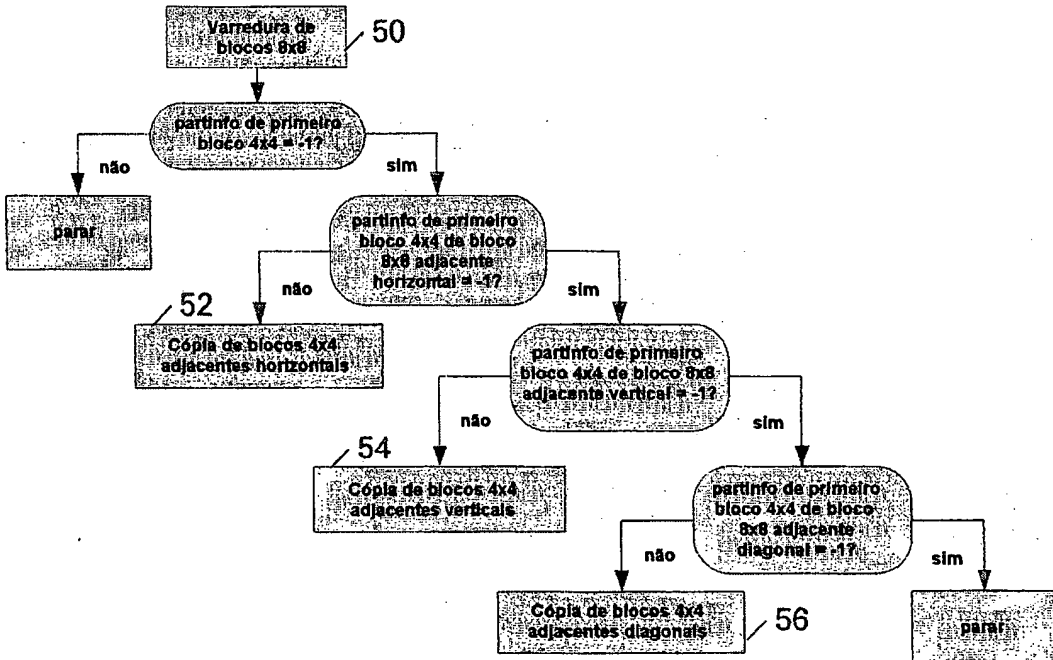


Figura 5

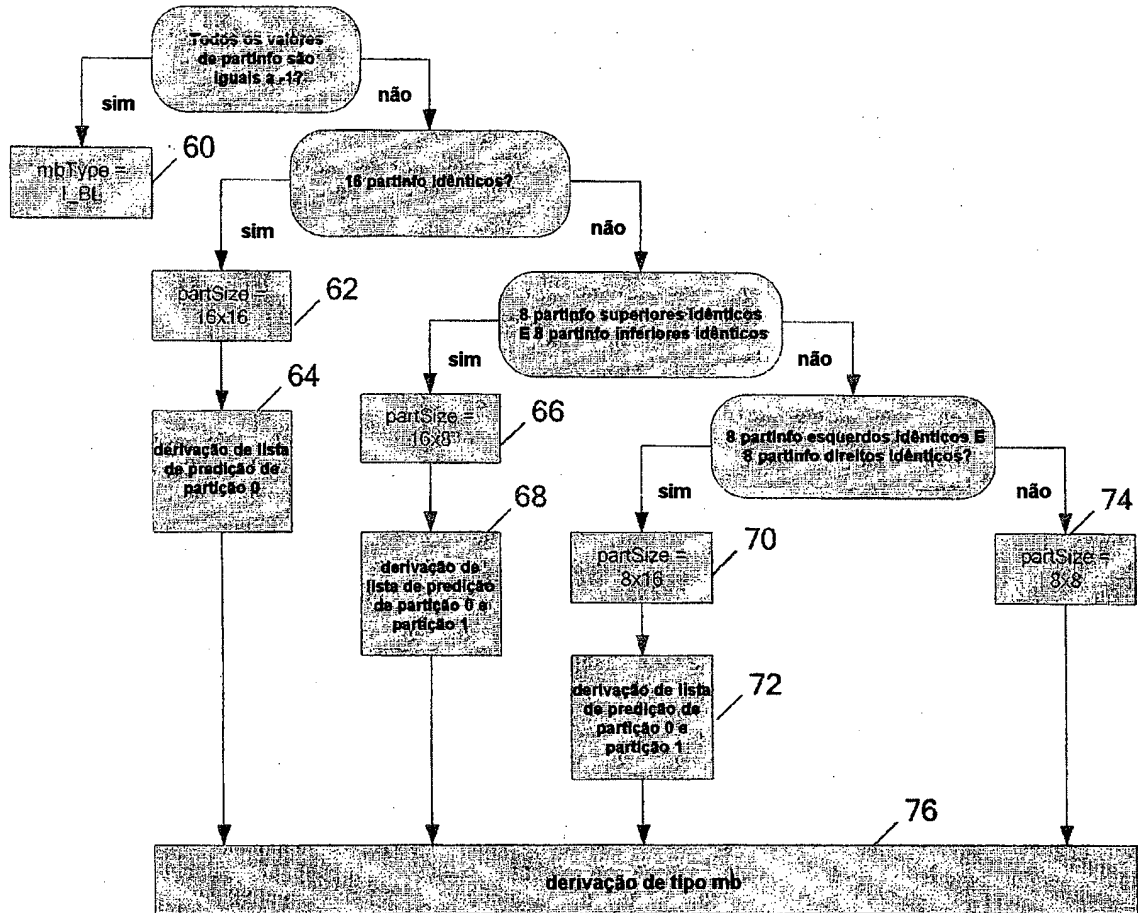


Figura 6

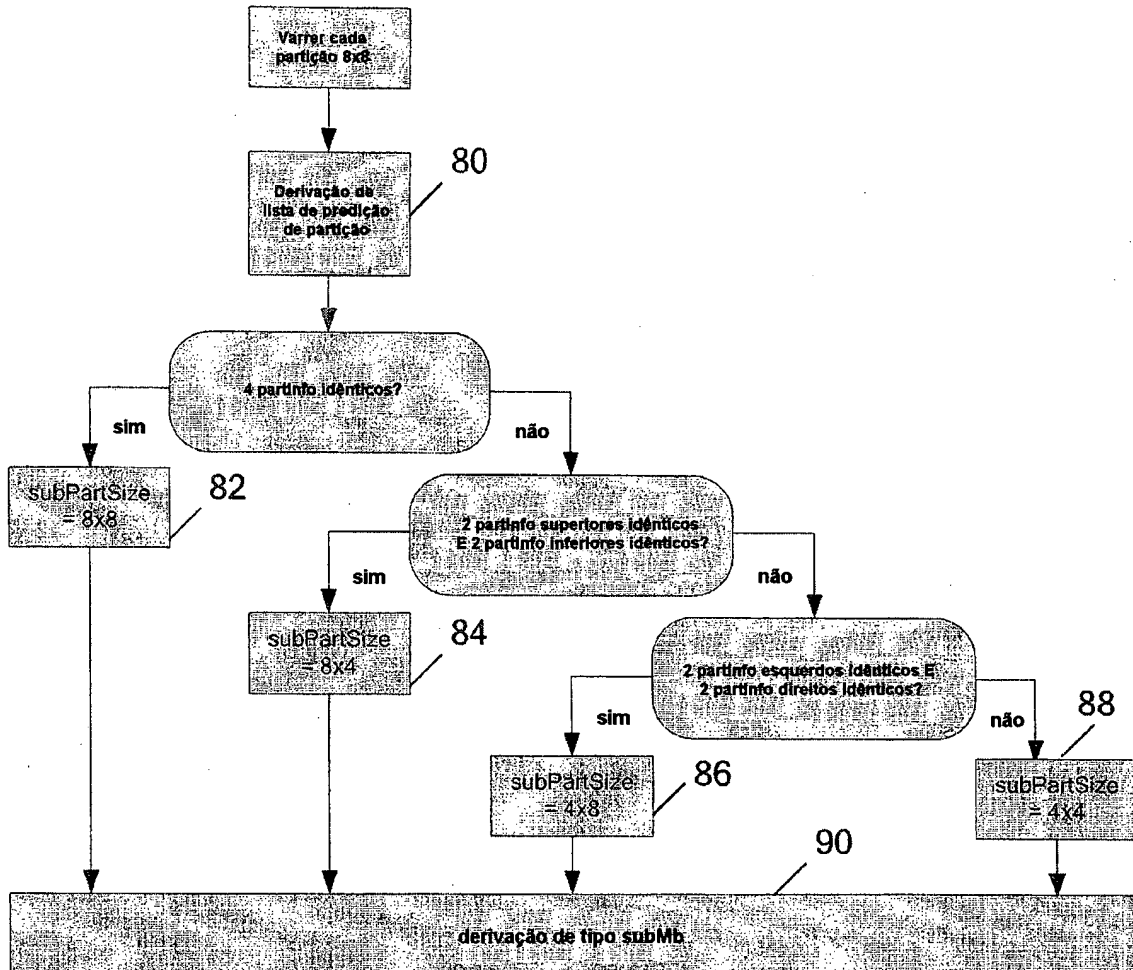


Figura 7

RESUMO

"PREDIÇÃO DE MOVIMENTO ENTRE CAMADAS SIMPLIFICADAS PARA CODIFICAÇÃO DE VÍDEO ESCALONÁVEL"

Trata-se de um método para determinar a um macrobloco de alta camada que usa
5 predição entre camadas um particionamento do macrobloco em partições. Esse
compreende as seguintes etapas: - dividir o macrobloco de camada alta em blocos de
camada alta não-sobrepostos de um tamanho predefinido; - determinar um pixel de camada
base correspondente para um pixel, chamado pixel de referência, de cada bloco de camada
alta; - identificar, para cada pixel de referência, um macrobloco de camada base ao qual o
10 pixel de camada correspondente pertence, uma partição de camada base à qual o dito pixel
de camada base correspondente no macrobloco de camada base identificado, uma
subpartição de camada base à qual o pixel de camada base correspondente pertence na
dita partição de camada base identificada se existir a dita subpartição; - derivar, para cada
bloco de camada alta, um único valor, chamado valor PartInfo, e - determinar um
15 particionamento do macrobloco de camada alta em partições de macrobloco ao comparar
entre esses os valores de PartInfo associados com cada bloco de alta camada.