



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

(11) PI 0502092-1 B1

(22) Data do Depósito: 13/04/2005

(45) Data de Concessão: 26/12/2017



(54) Título: "MÉTODO PARA CODIFICAÇÃO SEM PERDAS PREDITIVA DE MÍDIA DE IMAGEM E VÍDEO, MÉTODO DE DECODIFICAÇÃO DE DADOS CODIFICADOS SEM PERDAS PREDITIVOS DE UMA IMAGEM OU VÍDEO, E DECODIFICADOR DE VÍDEO OU IMAGEM CODIFICADOS"

(51) Int.Cl.: H04N 19/105; H04N 19/13; H04N 19/176; H04N 19/593; H04N 19/46; H04N 19/14

(52) CPC: H04N 19/105,H04N 19/13,H04N 19/176,H04N 19/593,H04N 19/46,H04N 19/14

(30) Prioridade Unionista: 15/04/2004 US 10/826.842

(73) Titular(es): MICROSOFT TECHNOLOGY LICENSING, LLC

(72) Inventor(es): KUNAL MUKERJEE

Relatório Descritivo da Patente de Invenção para
**"MÉTODO PARA CODIFICAÇÃO SEM PERDAS PREDITIVA DE MÍDIA DE
IMAGEM E VÍDEO, MÉTODO DE DECODIFICAÇÃO DE DADOS CODIFI-
CADOS SEM PERDAS PREDITIVOS DE UMA IMAGEM OU VÍDEO, E DE-
5 CODIFICADOR DE VÍDEO OU IMAGEM CODIFICADOS".**

CAMPO TÉCNICO

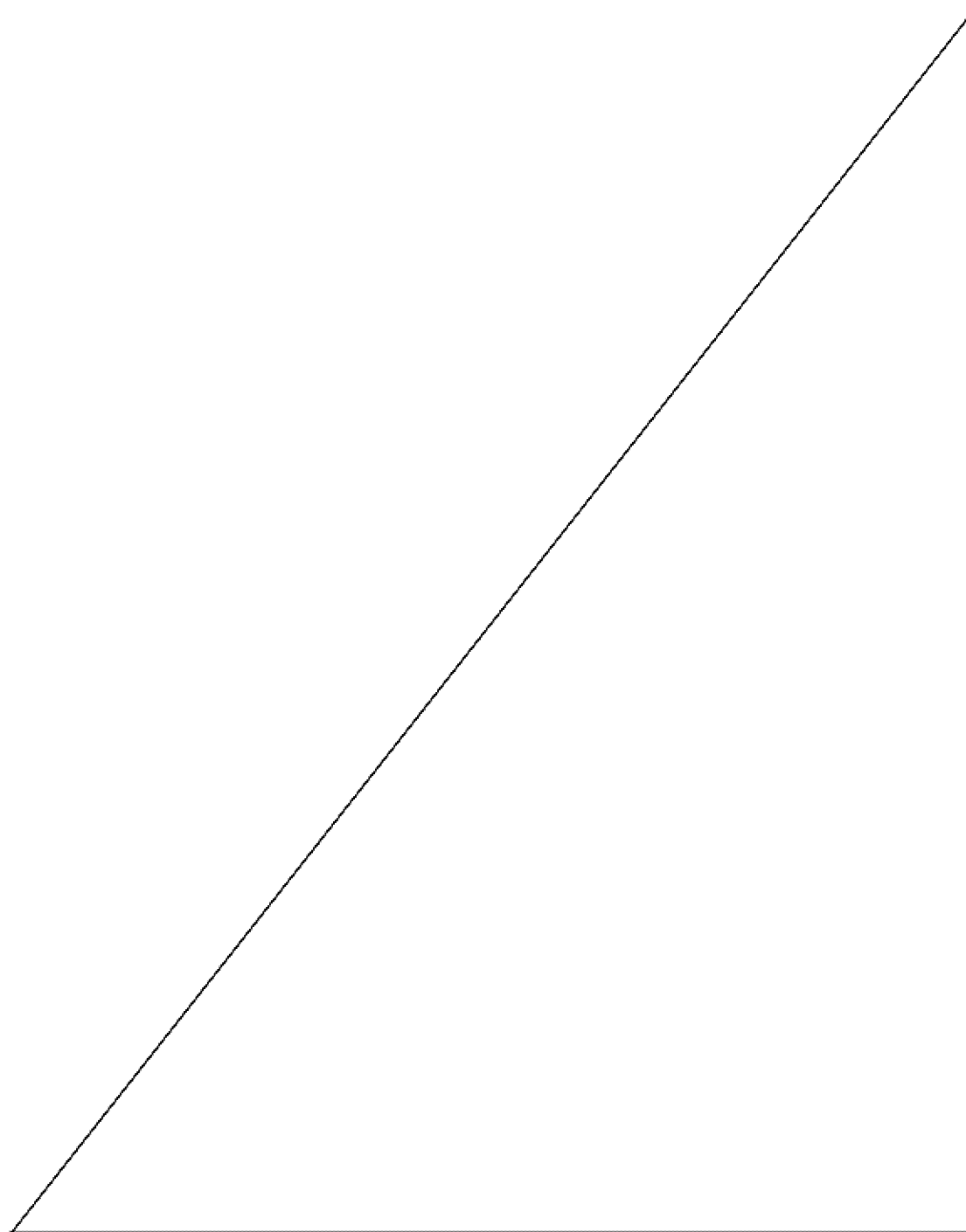
A invenção se refere à codificação à prova de perda de imagens e vídeo.

ANTECEDENTES

10 A codificação de imagens à prova de perda tem uma grande variedade de aplicações importantes, incluindo fotografia, filmografia, gráficos, etc. de alta qualidade. Também se aplica a codificação de vídeo de grau profissional, para codificar quadros de vídeo no ajuste de
15 qualidade mais alta possível, isto é, sem perdas. As imagens nessas aplicações podem ter diversas características, que apresenta um difícil desafio para elaborar um codificador / decodificador (codec) de imagem para ser aplicável genericamente por essas aplicações. Por exemplo,
20 plo, as imagens em gráficos têm, tipicamente, bordas pronunciadas ou de transições em cor (por exemplo, entre as cores do texto e de fundo, e nas bordas das formas adjacentes), enquanto que as imagens fotográficas são geralmente de tom contínuo (isto é, variam de forma constante em cor (por exemplo, como um gradiente) pela imagem).
25

Devido às diferenças nas características da imagem, os codecs de imagem mais genéricos são projetados para comprimir imagens fotográficas (por exemplo,

JPEG) ou gráficas (GIF). A compressão de imagem foto-
gráfica usa geralmente uma transformação de descorrela-
ção, como DCT ou ondas pequenas, enquanto que a com-
pressão gráfica usa, tipicamente, codecs à base de
5 cadeia, tal como LZ77 ou LZ78. Em geral, os



codecs fotográficos não funcionam bem em gráficos, por causa da consideração básica de que o nivelamento local ou polarização em corrente contínua, que fundamenta os métodos de transformação, é quebrado usualmente em gráficos. Contrariamente, os codecs gráficos fazem isso de forma pobre em imagens fotográficas, porque o alfabeto é muito grande para construir um bom dicionário. Por conseguinte, os codecs das imagens existentes para as imagens fotográficas não são projetados para fácil interoperabilidade com codecs de imagem e de vídeo, nem tratam o conteúdo gráfico de uma forma eficiente.

Por exemplo, CALIC (como descrito por X. Wu, N. Memon e K. Sayood, "A Context-Based Adaptative Lossless/Nearly-Lossless Coding Scheme For Continuous-Tone Images", ISO, 1995), JPEG-LS (como descrito por M. J. Weinberger e G. Serousi, "The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS", IEEE Trans. Image Processing., vol. 9, pp. 1.309 - 1.324, agosto de 2000) e SPIHT (como descrito por Said e W. A. Pearlman, "A New Fast And Efficient Image Codec Based On Set Partitioning In Hierarchical Trees", IEEE Trans. On Circuits and Systems for Video Technology, vol. 6, nº 6, pp. 243 - 250, junho de 1996) são codecs de imagem à prova de perda do estado da técnica atual para imagens fotográficas. No entanto, não são projetados para fácil interoperabilidade com os codecs de imagem e vídeo, nem tratam conteúdo gráfico eficientemente. Por outro lado, GIF é um codec gráfico à prova de perda do estado da técnica atual. Mas, também não trata o conteúdo

fotográfico, nem é fácil de incorporar dentro dele um codec de vídeo. PTC (como descrito por H. S. Malvar, "Fast Progressive Image Coding Without Wavelets", pp. 243 - 252, DCC 2000) é um codec à base de macrobloco, que pode ser facilmente integrado em códigos de imagem e/ou vídeo. No entanto, funciona muito bem com conteúdo gráfico. Além disso, BTPC (descrito por J. A. Robinson, "Efficient General-Purpose Image Compression With Binary Tree Predictive Coding", IEEE Transactions on Image Processing, vol. 6, nº 4, abril de 1997) é elaborado para tratar imagens fotográficas e gráficas em um projeto unificado e de velocidade otimizada, mas a sua compressão é longe da adequada.

SUMÁRIO

A codificação à prova de perda preditiva (PLC) aborda o problema mencionado acima de proporcionar compressão de imagem à prova de perda, que é geralmente aplicável a uma ampla variedade de imagens e vídeos. Uma implementação de PLC aqui ilustrada proporciona compressão de imagem à prova de perda de uma ampla gama de conteúdo fotográfico (imagem, vídeo e gráficos assemelhados) com eficiência de compressão comparável com os codecs de imagens à prova de perda existentes e execução em uma complexidade de tempo de realização mais rápido do que a maior parte dos codecs de imagens à prova de perda.

Várias implementações de codificação à prova de perda preditiva aqui descritas atingem esses resultados, por meio de uma combinação de pelo menos alguns dos seguintes pontos:

1. Opera no espaço de cores YCoCg. Esse espaço de cores aperfeiçoa a eficiência da codificação de conteúdo de imagens fotográficas e gráficas, e ainda uma conversão do espaço de cores para / da representação colorida RGB (vermelho - verde - azul) predominante pode ser feita por uso de
 5 um procedimento inteiramente para números inteiros, rápido.

2. Opera em macroblocos. Isso garante que a codificação à prova de perda preditiva possa ser facilmente integrada nos codecs de imagem e vídeo existentes, e é útil
 10 para as implementações de espaço / tempo escalonáveis em hardware e software, usando, por exemplo, segmentos.

3. Usa um rico conjunto de predições de modulação por códigos de pulsos diferenciais (DPCM), locais no nível de macrobloco. São projetados para descorrelacionar otimamente os dados de imagens de fontes fotográficas, bem como gráficas, sem recorrer a amostragens anormais ou reescaneamento da imagem (cf., J. A. Robinson, "Efficient General-Purpose Image Compression With Binary Tree Predictive Coding", IEEE Transactions on Image Processing, vol. 6, nº 4, abril de 1997; e X. Wu, N. Memon e K. Sayood, "A Context-Based Adaptative Lossless/Nearly-Lossless Coding Scheme For Continuous-Tone Images", ISO, 1995). As DPCMs são também projetadas para produzir resíduos que têm distribuições laplacianas de dois lados de polarização zero, pois esses são
 20 melhor codificados pelo método de codificação por entropia de Golomb Rice de densidade dupla (RLGR).

4. Usa o método de Golomb Rice de densidade dupla (RLGR) para codificar por entropia as várias distribuições

de símbolos.

Em uma implementação empregando essa combinação de pontos, a codificação à prova de perda preditiva proporciona compressão de todo o conteúdo fotográfico (imagem, vídeo e gráficos assemelhados), que não é apenas igual aquele do CALIC e maior do que os formatos existentes, incluindo JPEG-LS, PTC, BTSC, etc., mas também resulta em uma compressão igual ou superior a duas vezes para a maior parte do conteúdo gráfico, comparados com esses codecs do tipo fotográfico, e oferece uma complexidade de tempo de execução que é duas vezes ou mais mais rápida do que a maior parte dos codecs do tipo fotográfico de estado da técnica avançado.

Aspectos e vantagens adicionais da invenção vão ser tornados evidentes da seguinte descrição detalhada das modalidades, que é feita com referência aos desenhos em anexo.

BREVE DESCRIÇÃO DOS DESENHOS

A Figura 1 é um diagrama de blocos de um codificador de imagem utilizando codificação à prova de perda preditiva.

A Figura 2 é um diagrama ilustrando uma estrutura de macrobloco e segmentos da codificação à prova de perda preditiva utilizada no codificador da Figura 1.

A Figura 3 é um diagrama mostrando a vizinhança para um pixel no macrobloco, no qual a predição DPCM no codificador da Figura 1 é baseada.

A Figura 4 é um diagrama que ilustra os modos de predição DPCM empregados na codificação à prova de perda preditiva no codificador da Figura 1.

A Figura 5 é uma listagem de pseudo-códigos do procedimento de codificação para a codificação à prova de perda preditiva utilizada no codificador da Figura 1.

A Figura 6 é um diagrama de blocos de um decodificador de imagem utilizando codificação à prova de perda preditiva.

A Figura 7 é um diagrama de blocos de um meio físico de computação adequado para implementar o codec PLC das Figuras 1 e 6.

10 DESCRIÇÃO DETALHADA

A descrição apresentada a seguir é dirigida a implementações de codificação à prova de perda preditiva, que combina uma mistura de alguns ou todos da codificação por entropia Golomb Rice de densidade dupla (RLGB), modos DPCM
15 múltiplos, o espaço de cores YCoCg e uma estrutura de codificação de macroblocos (MB), para proporcionar um codec rápido e eficiente aplicável a uma ampla gama de conteúdo de imagem, incluindo fotográfico (tom contínuo), gráfico e de vídeo.

20 1. Codificador PLC

Com referência agora à Figura 1, um exemplo ilustrativo de um codificador de imagem 100, que é baseado na codificação à prova de perda preditiva (PLC), executa codificação ou compressão de dados de imagem 105. Os dados de
25 imagem, que são introduzidos no codificador PLC, podem ser quaisquer dos vários formatos de dados de imagem não comprimidos. Por exemplo, um formato comum processado pelo codificador de imagem ilustrado é o de dados de imagem vermelho -

verde - azul (RGB), tal como para uma imagem fotográfica ou gráfica, um quadro de vídeo, etc. Esses dados de imagem RGB são geralmente estruturados como um arranjo bidimensional de elementos de imagem (pixels), em que cada pixel é representado como uma amostra colorida vermelho - verde - azul (RGB) da imagem. Implementações alternativas do codificador de imagem podem usar outros formatos de dados de imagem de entrada. Deve-se reconhecer ainda que esse codificador pode ser incorporado dentro de um codificador de vídeo, para codificar um quadro dentro de uma sequência de vídeo, usando a codificação à prova de perda preditiva.

O codificador de imagem PLC 100 processa dos dados de imagem por meio de um conjunto de processos, que incluem um conversor de espaço de cores 110, um divisor de macrobloco 120, um modulador DPCM 130 e um codificador por entropia RLGR 140. O conversor de espaço de cores 110 converte pixels dos dados de imagem de entrada 105, de uma representação do espaço de cores exibível no espaço de cores YCoCg 115, o que aperfeiçoa a eficiência de codificação. O divisor de macroblocos 120 divide a imagem em macroblocos 125. O codificador DPCM 130 seleciona e aplica um de um conjunto de modos de predição DPCM disponíveis em cada macrobloco individual, que produz resíduos de predição tendo uma distribuição adequada para codificação por entropia RGR. O codificador por entropia RLGR 140 então codifica os resíduos da predição do macrobloco. Isso produz uma representação codificada por PLC dos dados de imagem.

1.1 Conversor do espaço de cores YCoCg

Mais especificamente, o conversor do espaço de cores 110 converte o formato colorido dos dados de imagem de entrada 105 para o espaço de cores YCoCg 115. Os pixels dos dados de imagem de entrada 105 são tipicamente representados em um formato de espaço de cores facilmente exibível, tal como o espaço de cores vermelho - verde - azul (RGB). O espaço de cores YCoCg 115 é mais adequado para codificação de imagem eficiente em espaço. Verificou-se que o YCoCg funciona bem para ambas as imagens fotográficas e gráficas, e supera outras transformações coloridas em termos de ganho de codificação. Mais especificamente, o espaço de cores à prova de perda YCoCg proporcionou um aperfeiçoamento de ~ 15%, comparado com o espaço de cores RGB na codificação PLC.

A conversão de espaço de cores RGB em YCoCg é feita antes de qualquer codificação dos dados de imagem de entrada de formato RGB. Nessa implementação ilustrada do codificador PLC 100, o conversor de espaço de cores 110 usa o processo de conversão de RGB em YCoCg descrito em mais detalhes por H. S. Malvar e G. J. Sullivan, "YCoCg-R: A Color Space With RGB Reversibility and Low Dynamic Range", Joint Video Team of ISO/IEC MPEG & ITU-T VCEG Doc. JVT-1014, julho de 2003, que proporciona um modo de descorrelacionar sem perda espaço de cores RGB em YCoCg. A conversão de espaço de cores RGB em YCoCg pode ser feita pela transformada avançada definida na seguinte equação (1).

$$\begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

O decodificador 600 (Figura 6) pode incluir uma reconversão para o espaço de cores RGB. Essa conversão usa a operação inversa definida na equação (2).

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 0 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix}$$

Etapas de afastamento são usadas para obter ambas as transformações avançada e oposta, resultando em procedimentos de conversão para números inteiros rápidos. As etapas de afastamento na direção avançada são dadas por:

$$Co = R - B;$$

$$x = B + (Co/2);$$

$$10 \quad Cg = G - x; \text{ e}$$

$$Y = x + (Cg/2).$$

É na direção inversa por:

$$x = Y - (Cg/2);$$

$$G = x + Cg;$$

$$15 \quad B = x - (Co/2); \text{ e}$$

$$R = Co + B.$$

Como pode-se notar, essas etapas de afastamento podem ser implementadas com todas as adições, subtrações e deslocamentos de bits de números inteiros, todos sendo extremamente rápidos.

20

1.2 Codificação de macroblocos

O divisor de macroblocos 120 divide a imagem em macroblocos (MBs) 125, como ilustrado na Figura 2. Em uma implementação do codificador de imagem 100, cada MB 125 é de um tamanho de 16 x 16 pixels. Implementações alternativas podem usar outros tamanhos de macroblocos. Essa estrutura de macroblocos pode ser prática e fácil de plugar o codificador de imagem à base de PLC em imagem popular, bem como codecs de vídeo. Também permite que implementações de hardware e software usem facilmente codificação de segmentos, nas quais o fluxo de bits codificado é empacotado em segmentos. Os segmentos são tipicamente algum número inteiro de fileiras de MBs. Isso produz uma área de cobertura de memória flexível e proporciona ela mesma uma capacidade de escalonamento de espaço / tempo (por exemplo, por uso de unidades de execução paralelas ou multiencadeada).

O divisor de macroblocos divide cada plano do espaço de cores YCoCg 115 em MBs 125 e codifica-os separadamente. Com cada MB individual, o codificador de imagem PLC 100 codifica os seguintes elementos de sintaxe: um modo DPCM 135, um modo MB 134 e os resíduos DPCM 136.

O elemento de modo DPCM identifica o modo DPCM 135 selecionado pelo modulador DPCM 130, para descorrelacionar os dados nesse MB 125. Nessa implementação, o modulador DPCM seleciona de oito modos DPCM possíveis, embora menos ou mais modos DPCM possam ser usados em implementações de codificador PLC alternativas. O divisor de macroblocos usa um contexto RLGR separado para codificar o modo DPCM para o MB.

Para o elemento de modo MB, o divisor de macroblo-
cos usa um alfabeto de dois símbolos, com um contexto de co-
dificação RLGB dedicado separado. O modo MB 134 sinaliza um
dos seguintes dois eventos: a) o MB codifica os valores
5 DPCM; ou b) o MB é "plano" e é portanto omitido. Neste caso,
o evento é tratado como uma saída precoce da codificação /
decodificação dos pixels do MB como resíduos DPCM 136, pois
o elemento de modo DPCM é suficiente para regenerar todos os
valores de pixels. Essa codificação de modo de macrobloco
10 plano é descrita em mais detalhes abaixo.

Finalmente, se o MB não for omitido, os resíduos
DPCM 136 são codificados no segmento MB do fluxo de saída
codificado por PLC.

1.3 Modos DPCM

15 O modulador DPCM 130 seleciona e aplica um modo
DPCM 135 para o MB corrente 125 que descorrelaciona mais o-
timamente o MB, para produzir resíduos DPCM 136 que comprí-
mem melhor com codificação por entropia RLGR. A codificação
por entropia RLGR atinge o seu melhor desempenho de codifi-
20 cação quando os seus valores de entrada têm uma distribuição
laplaciana de dois lados de polarização zero. Por exemplo, o
modulador DPCM 130, no codificador à base de PLC ilustrado,
pode comutar entre oito diferentes modos DPCM para descorre-
lacionar cada MB. Implementações alternativas podem incluir
25 menos ou mais modos DPCM. Esses vários diferentes modos DPCM
são elaborados para produzir resíduos tendo essa distribui-
ção ótima para vários diferentes modelos de pixels de MBs.
Isso propicia diferentes direções de bordas dominantes que

podem cair dentro do MB, para que sejam codificadas eficientemente.

Com referência à Figura 3, os vários modos DPCM especificam quais dos vizinhos de um pixel 300 são usados para prever o valor do pixel. Mais especificamente, o valor de cada pixel 310 do macrobloco é predito de alguma combinação de um ou mais pixels vizinhos 320 - 323. A diferença obtida da subtração de um valor efetivo do pixel do seu valor predito é o valor residual DPCM daquele pixel. Os oito modos DPCM no modulador DPCM ilustrado 130 usam as previsões baseadas nas combinações do pixel esquerdo vizinho 320, pixel esquerdo de topo 321, pixel de topo 322 e pixel direito de topo 323. Isso permite que o modo DPCM seja aplicado de uma passagem ou varredura de uma via, única pelo MB (isto é, varredura de cada fileira de pixels da esquerda para a direita, das fileiras de topo para o fundo do MB).

A Figura 4 ilustra os oito modos de predição DPCM usados no modulador DPCM ilustrado 130. Esses modos são elaborados para descorrelacionar os vários modelos de pixels comuns em um conjunto de resíduos tendo uma distribuição de símbolos que é melhor adequada para codificação por entropia RLGR. Mais especificamente, esses modos de predição DPCM são os seguintes:

Modo 0: Esse é o modo "bruto" ou sem DPCM, no qual os pixels individuais são codificados diretamente sem qualquer subtração. Esse modo é útil para MBs do tipo aleatório ou "ponto", sem qualquer predição consistentemente boa pelo MB.

Modo 1: Nesse modo DPCM, o valor do pixel é subtraído do seu vizinho esquerdo imediato, antes da codificação. Esse modo é útil quando grandes bordas se estendem ao longo da horizontal.

3 Modo 2: Neste modo DPCM, o valor do pixel é subtraído do seu vizinho de topo imediato. Esse modo é útil, quando as grandes bordas se estendem ao longo da vertical.

10 Modo 3: Neste modo DPCM, o valor é subtraído do mínimo dos seus vizinhos da esquerda e do topo, ou alternativamente do máximo dos seus vizinhos da esquerda e do topo. Esse modo DPCM é útil para as bordas diagonais de rampa que passam pela posição do pixel corrente.

15 Modo 4: Neste modo DPCM, o valor é subtraído da média dos seus vizinhos do topo e da direita. Esse modo DPCM é útil para as bordas da rampa diagonal com uma diferente orientação.

Modo 5: Neste modo DPCM, o valor é subtraído do seu vizinho da esquerda de topo. Esse valor DPCM é útil para bandas diagonais, por exemplo, em teor gráfico.

20 Modo 6: Neste modo DPCM, o modulador 130 subtrai o mesmo valor do seu vizinho esquerdo, como a diferença entre os seus vizinhos de topo e da esquerda de topo. Esse modo é útil para rampas horizontais em bandas.

25 Modo 7: Neste modo DPCM, o modulador 130 subtrai a média dos vizinhos da esquerda e de topo. Esse é também útil quando bordas diagonais dominam no MB.

O modulador DPCM testa cada um dos modos de predição DPCM 1 a 7 (isto é, outro modo diferente do DPCM, que é

o modo 0), de modo a selecionar que modo DPCM produz resíduos DPCM mais compressíveis. O modulador DPCM aplica os respectivos modos DPCM e checa a distribuição de símbolos dos resíduos resultantes. O modulador DPCM então checa que

5 modo de predição produziu resíduos tendo uma distribuição mais próxima à distribuição ideal para a codificação por entropia RLGR. O modulador DPCM checa ainda se essa distribuição mais próxima é suficientemente próxima à distribuição laplaciana de dois lados de polarização zero ideal. O modu-

10 lador DPCM seleciona o modo de predição DPCM com a distribuição mais próxima da ideal para o macrobloco, a menos que um limiar de suficiência não seja encontrado. De outro modo, se o modo de predição DPCM com a distribuição mais próxima da ideal não encontra o limiar de suficiência, então o modulador DPCM seleciona o modo sem DPCM (modo 0) como um padrão.

15

1.4 Codificação de modo MB plano

Como discutido acima, o codificador à base de PLC 100 também pode codificar um MB 125 no modo MB plano. O modo MB plano é usado quando os resíduos de DPCM, resultantes da

20 aplicação de um modo DPCM para o MB, são todos zero. Quando do teste dos modos DPCM, para selecionar o modo DPCM para uso no MB, o modulador DPCM checa ainda se o modo DPCM atualmente testado produz todos os resíduos de DPCM zero para o MB. Por determinação que um modo DPCM produz todos os resí-

25 duos de DPCM zero, o codificador à base de PLC 100 então codifica o MB no modo MB plano - sem precisar testar outros modos DPCM. Quando da codificação no modo MB plano, o codificador à base de PLC 100 pode codificar o MB no fluxo de

bits de saída, pois apenas o modo MB e o modo DPCM (isto é, a codificação de omissão dos resíduos de DPCM). A codificação do modo MB plano e do modo DPCM para o MB no fluxo de bits de saída é suficiente para decodificar os valores para os pixels dos MBs. Em virtude dos resíduos de DPCM não serem codificados, o modo MB plano produz uma maior eficiência de compressão para codificar o MB.

1.5 Contextos de Golomb Rice de densidade dupla (RLGR) múltiplos

Com referência novamente à Figura 1, o modo MB 134, o modo DPCM 135 e os resíduos de DPCM 136 produzidos pelo modulador DPCM 130 são depois codificados por entropia, usando codificação RLGR no codificador por entropia RLGR 140. O codificador por entropia RLGR no codificador à base de PLC ilustrado 100 usa o processo de codificação Golomb Rice de densidade dupla descrito por H. Malvar, "Fast Progressive wavelet coding", Proc. IEEE Data Compression Conference, Snowbird, UT, pp. 336 - 343, março - abril de 1999. A codificação RLGR faz a consideração de que o símbolo mais provável é zero. Desse modo, se uma cadeia de números com o símbolo mais provável não sendo zero for alimentado à codificação RLGR, vai ter um desempenho de codificação pobre. Se a sua entrada provém de uma fonte com distribuição de símbolos próxima da laplaciana, o processo de codificação RLGR vai codificar esses dados muito bem, muito estreitamente com a entropia, e em muitos casos vai fazer um trabalho melhor do que a codificação aritmética adaptativa. No codificador à base de PLC 100, os modos de predição DPCM são elaborados

para produzir distribuições laplacianas de dois lados de polarização zero de números inteiros assinados (para ambas as imagens fotográficas e gráficas) de conteúdos de imagens fotográficas e gráficas, nos quais o RLGR trabalha melhor.

5 O codificador por entropia RLGR 140 no codificador à base de PLC ilustrado usa um contexto RLGR separado para cada um de: a) o modo MB (plano ou não); b) o modo DPCM; c) os valores residuais DPCM (distribuições laplacianas de dois lados de polarização zero). Em cada um desses contextos, o

10 codificador por entropia RLGR executa uma codificação binária Golomb - Rice de densidade dupla adaptativa da cadeia binária formada pelos bits significativos que provêm dos símbolos sendo codificados pelo contexto separado, por exemplo, os resíduos DPCM para o contexto de valores residuais

15 DPCM. O uso de múltiplos contextos RLGR, para codificar diferentes distribuições de símbolos, aperfeiçoa o desempenho da codificação por entropia. Isso é porque é muito importante adaptar a cada distribuição individual e à sua distorção idiossincrática. Como um exemplo, os modos MB são mais pro-

20 váveis de serem todos planos. Consequentemente, a distribuição deles é provável de ser distorcida para longe do caso plano. Mas, isso pode virar ao contrário, se o conteúdo foi gráfico em vez de fotográfico. Dedicando-se um contexto RLGR específico, permite-se que o codificador por entropia RLGR

25 no codificador à base de PLC 100 se adapte a essas distribuições unimodais com maior eficiência. Nas implementações alternativas do codificador à base de PLC, mais ou menos contextos RLGR podem ser usados. O uso de contextos RLGR sepa-

5 rados adicionais codifiquem os resíduos DPCM nessas imple-
 mentações pode proporcionar ganhos de codificação por entropia maiores e impedir diluição do contexto, mas três contextos separados são usados na implementação ilustrada por razões práticas.

Após codificação por entropia pelo codificador por entropia RLGR, o multiplexador de fluxo de bits 150 monta os dados do codificador RLGR para o MB em um fluxo de bits de saída 195. Nas implementações usando codificação de segmentos, o multiplexador de fluxo de bits monta ou empacota os MBs codificados em segmentos.

1.6 Listagem de pseudo-códigos

A codificação PLC executada no codificador à base de PLC 100 é resumida na listagem de pseudo-códigos 500 mostrada na Figura 5. Nessa listagem de pseudo-códigos, o parâmetro de entrada "ImageBand" representa dados de imagem de uma das coordenadas do espaço de cores, isto é, Y, Co ou Cg. Esse processo de codificação PLC é invocado após o espaço de cores da imagem para o espaço de cores YCoCg.

20 2. Codificador PLC

Com referência agora à Figura 6, um decodificador de imagem 600, à base de codificação à prova de perda preditiva (PLC), executa decodificação do fluxo de bits de saída 195, produzido da codificação PLC pelo codificador de imagem 25 à base de PLC 100. Nesse decodificador de imagem à base de PLC 600, um demultiplexador de fluxo de bits 610 primeiro separa os MBs codificados individuais 125 no fluxo de bits, e o modo MB codificado 134, o modo DPCM 135 e os resíduos

DPCM 136 para aquele MB 125. O demultiplexador de fluxo de bits proporciona os dados separados para um decodificador RLGR 620.

O decodificador RLGR 620 decodifica o modo MB 134
 5 codificado por RLGR, o modo DPCM 135 e os resíduos DPCM 136 para cada MB 125. O decodificador RLGR 620 usa o processo de decodificação RLGR descrito por H. Malvar, "Fast Progressive wavelet coding", Proc. IEEE Data Compression Conference, Snowbird, UT, pp. 336 - 343, março - abril de 1999. O deco-
 10 dificador RLGR 620 então proporciona dados a um demodulador DPCM 630.

O demodulador DPCM 630 executa o processo inverso nos resíduos DPCM para o modo de predição DPCM, que foi usado para o MB, restaurando, desse modo, os dados do MB. Para
 15 um MB codificado no modo MB plano, o demodulador DPCM 630 executa o processo inverso para todos os resíduos zero para o modo de predição DPCM decodificado.

Após aplicação da predição DPCM inversa, um re-
 construtor de imagem 640 remonta os MBs para reconstruir a
 20 imagem 115. Um conversor de espaço de cores 640 executa depois o inverso da conversão do espaço de cores YCoCg, para converter esses dados de imagem de volta a uma imagem RGB 105. Em algumas implementações, essa conversão pode ser omitida e a imagem deixada no formato do espaço de cores YCoCg
 25 115.

3. Meio físico de computação

O codificador à base de PLC descrito acima 100 e/ou o decodificador 600 (codec PLC) pode(m) ser implementa-

do(s) em qualquer um de uma variedade de dispositivos de processamento de vídeo e dispositivos de computação, incluindo computadores de vários fatores de forma (pessoal, estação de trabalho, servidor, portátil, de colo, mesa digitalizadora ou outro móvel), redes de computação distribuídas e serviços da rede mundial de computadores, e gravadores / reprodutores / receptores / visualizadores de imagem e vídeo, como uns poucos exemplos genéricos. O codec à base de PLC pode ser implementado em um conjunto de circuitos de hardware, bem como em software de codec 780, por execução dentro de um computador ou outro meio físico de computação, tal como mostrado na Figura 7.

A Figura 7 ilustra um exemplo genérico de um dispositivo de processamento de imagem / vídeo adequado em um meio físico de computação 700 (por exemplo, de um computador), no qual as técnicas descritas podem ser implementadas. Esse meio físico 700 não é intencionado para sugerir qualquer limitação para o âmbito de uso ou funcionalidade da invenção, pois a presente invenção pode ser implementada em diversos meios físicos de processamento de imagem / vídeo multipropósito ou de finalidade específica.

Com referência à Figura 7, o meio físico de computação 700 inclui pelo menos uma unidade de processamento 710 e uma memória 720. Na Figura 7, essa configuração mais básica 730 é incluída dentro de uma linha pontilhada. A unidade de processamento 710 executa instruções executáveis por computador e pode ser um processador real ou virtual. Em um sistema multiprocessamento, unidades de processamento múlti-

plas executam instruções executáveis por computador, para aumentar o poder de processamento. A memória 720 pode ser memória volátil (por exemplo, registradores, cache, RAM), memória não volátil (por exemplo, ROM, EEPROM, memória instantânea, etc.) ou algumas combinações das duas. A memória 720 armazena software 780 implementando o codec à base de PLC.

Um meio físico de computação pode ter recursos adicionais. Por exemplo, o meio físico de computação 700 inclui armazenamento 740, um ou mais dispositivos de entrada 750, um ou mais dispositivos de saída 760 e uma ou mais conexões de comunicação 770. Um mecanismo de interconexão (não mostrado), tal como um barramento, um controlador ou rede, interliga os componentes do meio físico de computação 700. Tipicamente, o software do sistema operacional (não mostrado) proporciona um meio físico operacional para outro software em execução no meio físico de computação 700 e coordena as atividades dos componentes do meio físico de computação 700.

O armazenamento 740 pode ser removível ou não removível e inclui discos magnéticos, fitas ou cassetes magnéticos, CD-ROMs, CD-RWs, DVDs, ou qualquer outro meio que possa ser usado para armazenar informações e que pode ser acessado dentro do meio físico de computação 700. O armazenamento 740 armazena instruções para o software de codec à base de PLC 780.

O(s) dispositivo(s) de entrada 750 (por exemplo, para os dispositivos operando como um ponto de controle na arquitetura de conectividade do dispositivo 100) pode ser um

dispositivo de entrada de toque, tal como um teclado, um mouse, uma caneta ou um trackball, um dispositivo de entrada por voz, um dispositivo de escaneamento, ou um outro dispositivo que proporciona entrada para o meio físico de computação 700. Para áudio, o(s) dispositivo(s) de entrada 750 pode ser um cartão de som ou um dispositivo similar que aceita entrada de áudio em forma analógica ou digital, ou um leitor de CD-ROM, que proporciona amostras de áudio para o meio físico de computação. O(s) dispositivo(s) de saída 760 10 pode ser um visor, uma impressora, um alto-falante, um gravador de CD, ou um outro dispositivo que proporciona saída do meio físico de computação 700.

A ou as conexões de comunicação 770 propiciam comunicação por um meio de comunicação para outra entidade de 15 computação. O meio de comunicação transporta informações, tais como instruções executáveis por computador, informações de áudio / vídeo ou de outra mídia, ou outros dados em um sinal de dados modulado. Um sinal de dados modulado é um sinal que tem uma ou mais das suas características ajustadas 20 ou alteradas de uma maneira tal a codificar as informações no sinal. Por meio de exemplo, e não limitação, os meios de comunicação incluem técnicas ligadas ou sem fio implementadas com um veículo elétrico, óptico, de RF, infravermelho, acústico ou outro.

25 O processamento de macroexpansão e as técnicas de exibição da presente invenção podem ser descritas no contexto geral de meios legíveis por computador. Os meios legíveis por computador são quaisquer meios disponíveis que podem ser

acessados dentro de um meio físico de computação. Por meio de exemplo, e não limitação, com o meio físico de computação 700, os meios legíveis por computador incluem a memória 720, o armazenamento 740, os meios de comunicação e combinações de quaisquer dos mencionados acima.

As técnicas da presente invenção podem ser descritas no contexto geral de instruções executáveis por computador, tais como aquelas incluídas em módulos de programas, sendo executadas em um meio físico de computação em um processador real ou virtual alvo. Geralmente, os módulos de programas incluem rotinas, programas, bibliotecas, objetos, classes, componentes, estruturas de dados, etc., que executam tarefas particulares ou implementam tipos de dados abstratos particulares. A funcionalidade dos módulos de programas pode ser combinada ou dividida entre os módulos de programas, como desejado, nas várias modalidades. As instruções executáveis por computador para os módulos dos programas podem ser executadas dentro de um meio físico de computação local ou distribuído.

No interesse de apresentar, a descrição detalhada usa termos como "determinar", "gerar", "ajustar" e "aplicar" para descrever operações de computador em um meio físico de computação. Esses termos são abstrações de alto nível para as operações executadas por um computador e não devem ser confundidos com atos executados por um ser humano. As operações efetivas do computador correspondentes a esses termos variam, dependendo da implementação.

Em vista das muitas possíveis modalidades nas
quais podem ser aplicados os princípios da presente inven-
ção, a invenção vai ser reivindicada para essas modalidades,
como estando dentro do âmbito e do espírito das reivindica-
5 ções e equivalentes apresentados a seguir.

REIVINDICAÇÕES

1. Método para codificação sem perdas preditiva de mídia de imagem e vídeo caracterizado pelo fato de que compreende as etapas de:

5 dividir dados de imagem de entrada em macro-blocos;

 para um individual um dos macro-blocos, selecionar um modo dos múltiplos modos de predição de modulação de código de pulso diferencial disponíveis para aplicar
10 ao macro-bloco que, fora dos modos de predição de modulação de código de pulso diferencial disponíveis, produz uma Distribuição Laplaciana de dois lados de polarização zero próxima à ótima de uma codificação por entropia de Golomb Rice de densidade dupla;

15 aplicar o modo de predição de modulação de código de pulso diferencial selecionado ao macro-bloco;

 codificar por entropia de Golomb Rice de densidade dupla resíduos de modulação de código de pulso diferencial do macro-bloco;

20 em que selecionar o modo de predição de modulação de código de pulso diferencial compreende:

 determinar se o modo de predição de modulação de código de pulso diferencial que fornece a Distribuição de Laplace de dois lados de polarização zero próxima à ótima
25 de uma codificação por entropia de Golomb Rice de densidade dupla é mais próximo do que um limite de suficiência para a Distribuição de Laplace de dois lados de polarização zero ótima para a codificação por entropia de Golomb

Rice de densidade dupla; e

se não estiver mais próximo do referido limite de suficiência, aplicar nenhuma modulação de código de pulso diferencial no macro-bloco antes da codificação por entropia de Golomb Rice de densidade dupla.

2. Método, de acordo com a reivindicação 1, caracterizado pelo fato de que compreende converter os dados de imagem de entrada em um formato de espaço de cores YCoCg.

10 3. Método, de acordo com a reivindicação 2, caracterizado pelo fato de que a conversão no formato de espaço de cores YCoCg precede a divisão em macro-blocos.

4. Método, de acordo com a reivindicação 1, caracterizado pelo fato de que compreende ainda a codificação do modo de predição de modulação de código de pulso diferencial selecionada para o macro-bloco usando codificação de entropia de Golomb-Rice de densidade dupla.

5. Método de acordo com a reivindicação 1, caracterizado pelo fato de que compreende ainda a codificação do modo de predição de modulação de código de pulso diferencial e dos resíduos de modulação do código de pulso diferencial com contextos de codificação de Golomb-Rice de densidade dupla separados.

6. Método de acordo com a reivindicação 1, caracterizado pelo fato de que compreende ainda determinar se a aplicação do modo de predição de modulação de código de pulso diferencial selecionado ao macro-bloco produz todos os resíduos de modulação de código de pulso dife-

rencial de valor zero; e

codificar uma indicação de que o macro-bloco é plano, em vez de codificação por entropia de Golomb-Rice de densidade dupla de resíduos de modulação de código de pulso diferencial do macro-bloco.

7. Método, de acordo com a reivindicação 1, caracterizado pelo fato de que os modos de predição de modulação de código de pulso diferencial compreendem modos designados para produzir uma distribuição ótima de Laplace de dois lados de polarização zero para codificação por entropia de Golomb-Rice de densidade dupla para macro-blocos cujos conteúdos de imagem são predominantemente uma grande borda horizontal, uma grande borda vertical, bordas diagonais em rampa e rampas horizontais em banda

8. Método, de acordo com a reivindicação 1, caracterizado pelo fato de que os modos de predição de modulação de código de pulso diferencial compreendem:

um primeiro modo no qual um valor do pixel é subtraído do seu pixel vizinho esquerdo;

um segundo modo no qual um valor do pixel é subtraído do seu pixel vizinho de topo;

um terceiro modo no qual um valor do pixel é subtraído de um mínimo ou máximo dos seus pixels vizinhos esquerdo e de topo;

um quarto modo no qual um valor do pixel é subtraído de uma média dos seus pixels vizinhos de topo e da direita de topo;

um quinto modo no qual um valor do pixel é sub-

traído do seu pixel vizinho de topo à esquerda;

um sexto modo no qual a diferença entre um pixel de topo e os seus pixels vizinhos de topo à esquerda é subtraída do seu pixel vizinho da esquerda; e

5 um sétimo modo no qual um valor do pixel é subtraído de uma média dos pixels da esquerda e dos pixels vizinhos de topo.

9. Método de decodificação de dados codificados sem perdas preditivos de uma imagem ou vídeo, caracteri-
10 zado pelo fato de que compreende:

decodificar por entropia de Golomb-Rice de densidade dupla um modo de macro-bloco, um modo de predição de modulação de código de pulso diferencial e resíduos de modulação de código de pulso diferencial, para cada um de
15 uma pluralidade de macro-blocos usando contextos de codificação Golomb-Rice de densidade dupla separados,

em que, durante a codificação, o modo de predição de modulação de código de pulso diferencial foi selecionado de múltiplos modos de predição de modulação de
20 código de pulso diferencial disponíveis, o modo de predição de modulação de código de pulso diferencial produzindo uma distribuição residual do macro-bloco para fornecer a correspondência mais próxima a uma Distribuição de Laplace de dois lados de polarização zero ótima para uma
25 codificação por entropia de Golomb-Rice de densidade dupla;

onde o modo de macro-bloco de um macro-bloco é um modo de bloco de macro plano, decodificar os pixels do

macro-bloco usando uma demodulação de modulação de código de pulso diferencial que é um inverso do modo preditivo de modulação do código de pulso diferencial decodificado por Golomb-Rice de densidade dupla de todos os resíduos zero;

caso contrário, onde o modo de predição de modulação do código de pulso diferencial do macro-bloco é um modo de predição de modulação de código de pulso não diferencial, decodificar os pixels do macro-bloco sem demodulação de modulação do código de pulso diferencial;

caso contrário, demodular os resíduos de modulação de código de pulso diferencial decodificado pelo Golomb-Rice usando uma demodulação de modulação de código de pulso diferencial que é um inverso do modo de predição de modulação de código de pulso diferencial de Golomb-Rice decodificado de densidade dupla; e

montar os macro-blocos para formar dados de imagem decodificados.

10. Método de acordo com a reivindicação 9, caracterizado pelo fato de que compreende converter os dados de imagem decodificados de um formato de espaço de cor YCoCg para um formato de espaço de cor exibível.

11. Decodificador de vídeo ou imagem codificados sem perdas preditivo caracterizado pelo fato de que executa um método como definido na reivindicação 9 ou 10, compreendendo:

um decodificador de entropia Golomb-Rice de densidade dupla operando para decodificar os resíduos de

modulação de código de pulso diferencial codificado com Golomb-Rice de densidade dupla e o modo de predição de modulação de código de pulso diferencial de um macro-bloco;

5 um demodulador de modulação de código de pulso diferencial para aplicar um inverso do modo de predição de modulação de código de pulso diferencial aos resíduos de modulação do código de pulso diferencial; e

 um reassembler do macro-bloco para montar o ma-
10 cro-bloco com outros macro-blocos decodificados para formar dados de uma imagem reconstruída.

 12. Decodificador de acordo com a reivindicação
11, caracterizado pelo fato de que compreende um conversor YCoCg inverso para converter a imagem reconstruída de
15 um espaço de cores YCoCg para um espaço de cores adequado para exibir a imagem.

Figura 1

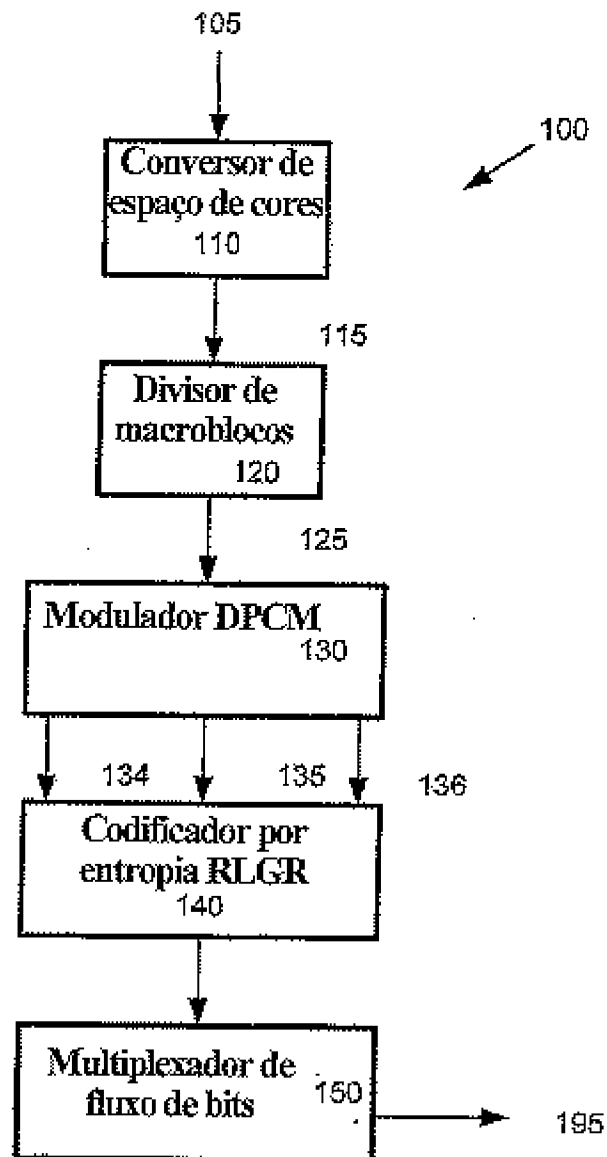


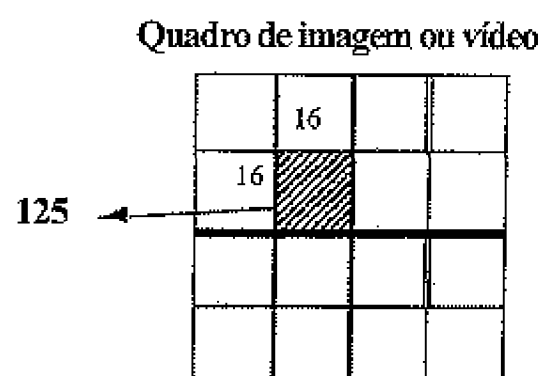
Figura 2

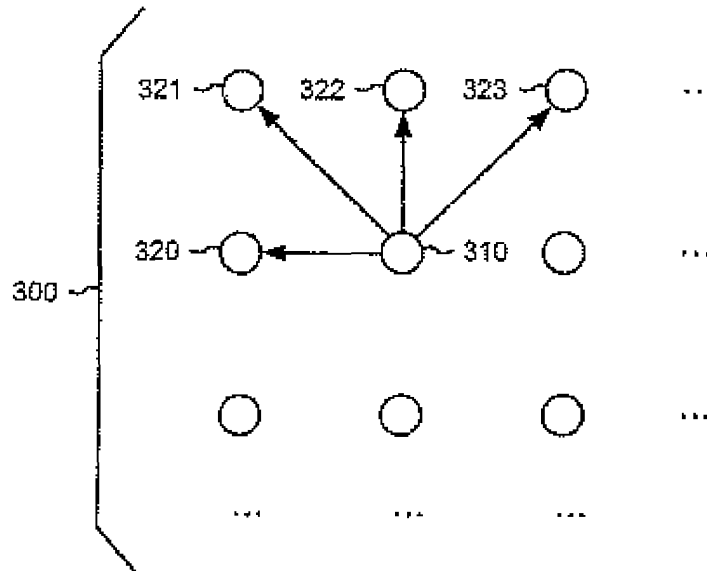
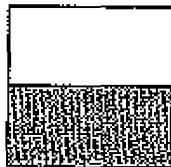
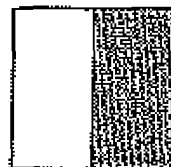
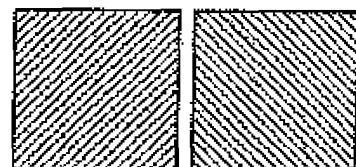
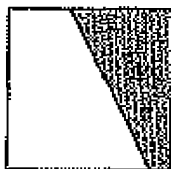
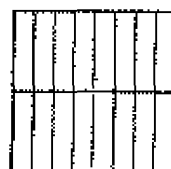
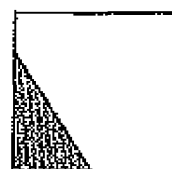
Figura 3**Figura 4****Modos DPCM****Modo 1****Modo 2****Modo 3****Modo 4****Modo 5****Modo 6****Modo 7**

Figura 5

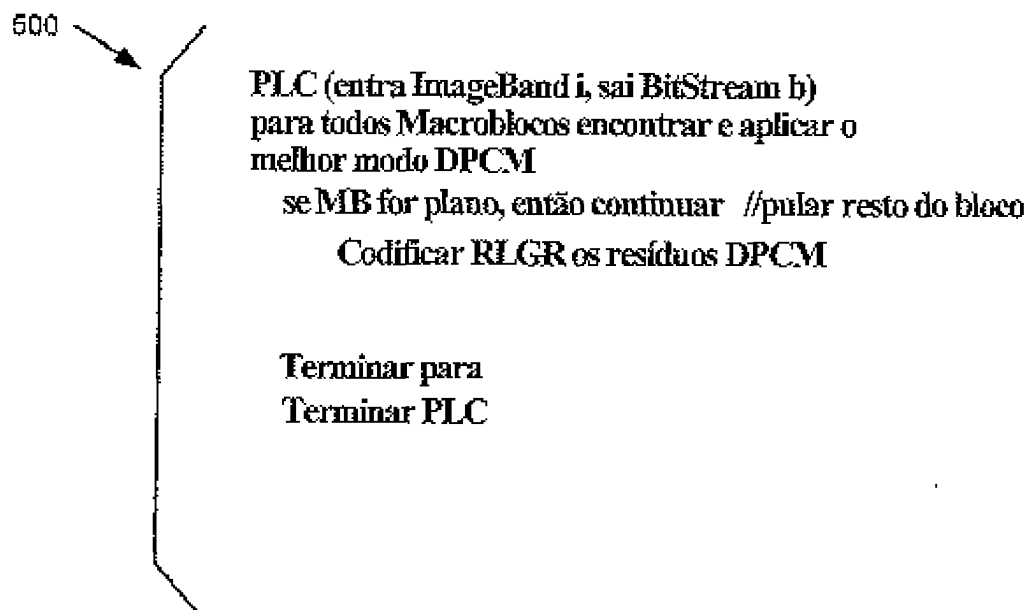


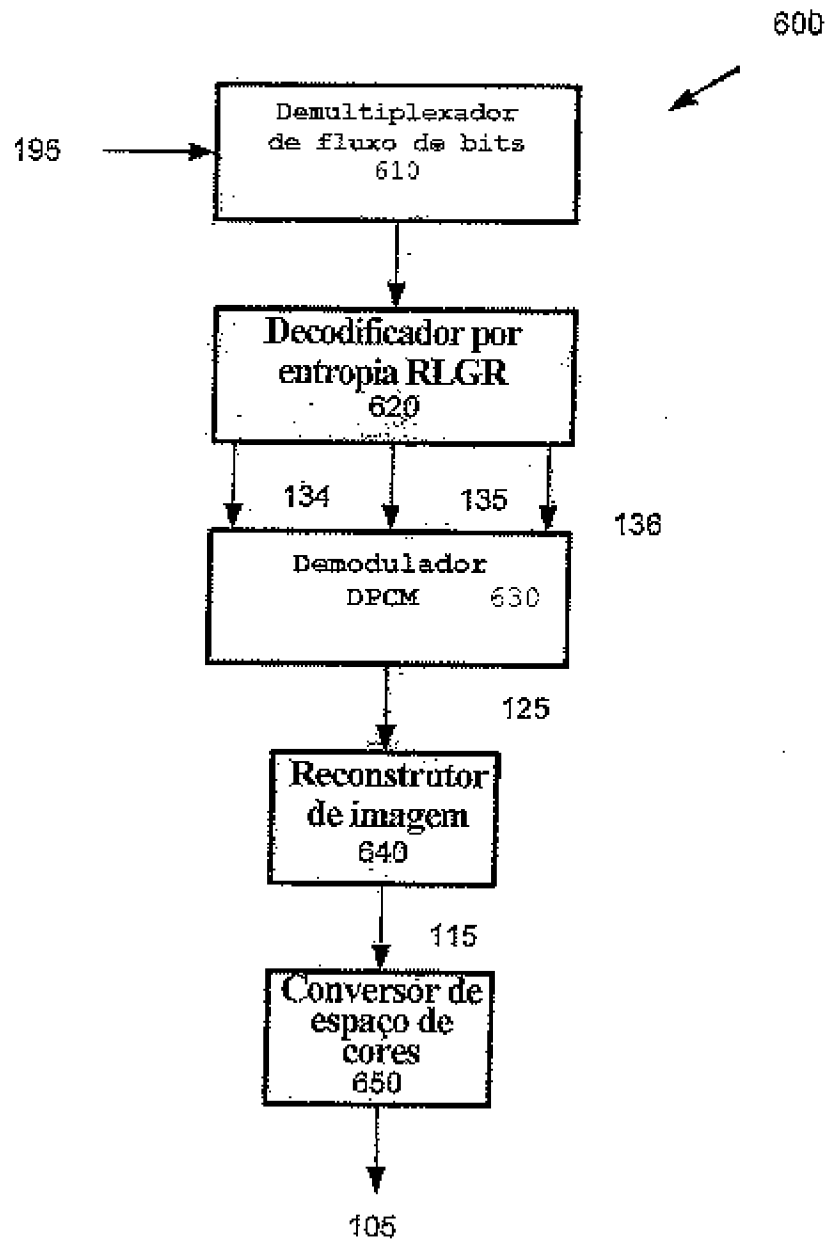
Figura 6

Figura 7