



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

(21) **PI0710704-8 A2**



* B R P I 0 7 1 0 7 0 4 A 2 *

(22) Data de Depósito: 04/05/2007
(43) Data da Publicação: 23/08/2011
(RPI 2120)

(51) *Int.Cl.:*
H04N 7/24 2011.01

(54) Título: **QUANTIZAÇÃO FLEXÍVEL**

(30) Prioridade Unionista: 05/05/2006 US 11/418,690

(73) Titular(es): Microsoft Corporation

(72) Inventor(es): Chengjie Tu, Sridhar Srinivasan

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

(86) Pedido Internacional: PCT US2007010848 de 04/05/2007

(87) Publicação Internacional: WO 2007/130580 de 15/11/2007

(57) **Resumo:** QUANTIZAÇÃO FLEXÍVEL Um codificador/decodificador de meios digitais utiliza uma técnica de quantização flexível que provê a capacidade de variar a quantização ao longo de várias dimensões dos dados de meios digitais codificados, incluindo espacial, subfaixas de frequência e canais de cor. O codec utiliza um esquema de sinalização para sinalizar várias permutações das combinações de quantização flexível eficientemente para cenários de utilização principal. Quando uma escolha do quantizador está disponível, o codec codifica eficientemente o quantizador atual mediante definição de um subconjunto de quantizadores e indexa o quantizador atual a partir do conjunto.

700

FRAME () {	Num bits
...	lots
DC_FRAME_UNIFORM	1
se (DC_FRAME_UNIFORM)	
DC_QUANTIZER	variável
se ('passa-baixa não-saltada')	
USE_DC_QUANTIZER	1
se (USE_DC_QUANTIZER == FALSE){	
LP_FRAME_UNIFORM	1
se (LP_FRAME_UNIFORM){	
LP_QUANTIZERS = 1	
LP_QUANTIZER()	variável
}	
}	
}	
se ('passa-alta não-saltada')	
USE_LP_QUANTIZER	1
se (USE_LP_QUANTIZER == FALSE){	
HP_FRAME_UNIFORM	1
se (HP_FRAME_UNIFORM){	
HP_QUANTIZERS = 1	
HP_QUANTIZER()	variável
}	
}	
}	

Antecedentes

1. Codificação por Transformada de Bloco

Codificação por transformada é uma técnica de compactação usada em muitos sistemas de compactação de áudio, imagem e vídeo. Imagem digital e vídeo não-compactados são representados tipicamente ou capturados como amostras de elementos de figura ou cores em locais em um quadro de imagem ou vídeo arranjado em uma grade bidimensional (2D). Isso é referido como representação de domínio espacial da imagem ou vídeo. Por exemplo, um formato típico para imagens consiste em um fluxo de amostras de elemento de figura em cores de 24 bits arranjadas como uma grade. Cada amostra é um número representando componentes de cor em um local de pixel na grade dentro de um espaço de cor, tal como RGB, ou YIQ, entre outros. Vários sistemas de imagem e vídeo podem usar várias resoluções de amostragem de cor, espacial e temporal, diferentes. Similarmente, áudio digital é representado tipicamente como fluxo de sinal de áudio amostrado temporalmente. Por exemplo, um formato de áudio típico consiste em um fluxo de amostras de amplitude de 16 bits de um sinal de áudio captado em intervalos regulares e tempo.

Sinais não-compactados digitais de áudio, imagem e vídeo, podem consumir considerável capacidade de armazenamento e transmissão. A codificação por transformada reduz o tamanho do áudio, imagens e vídeo, digitais, mediante transformação da representação de domínio-espacial do sinal em uma representação de domínio-freqüência (ou outro domínio por transformada semelhante) e, então, reduzindo a resolução de certos componentes de freqüência geralmente menos perceptíveis da representação de domínio-transformada. Isso geralmente produz degradação muito menos perceptível do sinal digital em comparação com a redução de resolução de cor ou espacial das imagens ou vídeos no domínio espacial, ou de áudio no domínio temporal.

Mais especificamente, um codec baseado em transformada de bloco típico 100 mostrado na Figura 1 divide os pixels não-compactados da imagem digital em dois blocos dimensionais de tamanho fixo (X_1, \dots, X_n), cada bloco possivelmente sobreposto a outros blocos. Uma transformada linear 120, 121 que realiza análise de freqüência-espacial é aplicada a cada bloco, a qual converte as amostras espaçadas dentro do bloco em um conjunto de coeficientes de freqüência (ou transformada) geralmente representando a intensidade do sinal digital nas bandas de freqüência correspondentes através do intervalo de blocos. Para compreensão, os coeficientes de transformada podem ser quantizados seletivamente 130 (isto é, reduzidos em resolução, tal como mediante descarte dos bits menos significativos dos valores de coeficiente ou de outro modo mapeando os valores em um número de resolução superior ajustado para uma resolução inferior), e também codificados com comprimento variável ou entropia 130 para um fluxo de dados compactado. Na decodificação, os coefi-

cientes da transformada transformação inversamente 170-171 para quase reconstruir a cor original/imagem amostrada espacial/sinal de vídeo (blocos reconstruídos $\hat{X}_1, \dots, \hat{X}_n$).

A transformada por blocos 120-121 pode ser definida como uma operação matemática em um vetor x de tamanho N . Mais freqüentemente, a operação é uma multiplicação linear, produzindo a saída de domínio de transformada $y = M x$, M sendo a matriz da transformada. Quando os dados de entrada são arbitrariamente longos, eles são segmentados em vetores de N tamanhos, e uma transformada por blocos é aplicada a cada segmento. Para a finalidade de compactação de dados, as transformadas de blocos reversíveis são escolhidas. Em outras palavras, a matriz M é reversível. Em múltiplas dimensões (por exemplo, para imagem e vídeo), as transformadas por blocos são implementadas tipicamente como operações separáveis. A multiplicação de matriz é aplicada separadamente ao longo de cada dimensão dos dados (isto é, ambas, fileiras e colunas).

Para compactação, os coeficientes da transformada (componentes de vetor y) podem ser seletivamente quantizados (isto é, reduzidos em resolução, tal como mediante descarte dos bits menos significativos dos valores de coeficiente ou de outro modo mapeando os valores em um número de resolução superior ajustado para uma resolução inferior), e também codificados em comprimento variável ou por entropia em um fluxo de dados compactado.

Na decodificação no decodificador 150, o inverso dessas operações (decodificação por entropia/desquantização 160 e transformada inversa por blocos 170-171) é aplicado pelo lado do decodificador 150, conforme mostrado na Figura 1. Enquanto reconstruindo os dados, a matriz inversa M^{-1} (transformada inversa 170-171) é aplicada como um multiplicador aos dados de domínio da transformada. Quando aplicada aos dados de domínio da transformada, a transformada inversa quase reconstrói os meios digitais de domínio temporal ou domínio espacial original.

Em muitas aplicações de codificação baseadas em transformada por blocos, a transformada é desejavelmente reversível para suportar ambas, compactação de perda e compactação sem perda dependendo do fator de quantização. Sem quantização (geralmente representado como um fator de quantização de 1), por exemplo, um codec utilizando uma transformada reversível pode reproduzir exatamente os dados de entrada na decodificação. Contudo, a exigência de reversibilidade nessas aplicações restringe a escolha das transformadas a partir das quais o codec pode ser projetado.

Muitos sistemas de compactação de imagem e vídeo, tal como MPWG e Windows Media, entre outros, utilizam transformadas com base na Transformada Discreta do Coseno (DCT). A DCT é conhecida como tendo propriedades favoráveis de compactação de energia e resultam em compactação de dados quase ótima. Nesses sistemas de compactação, a DCT inversa (IDCT) é empregada nos laços de reconstrução em ambos, no codifica-

dor e no decodificador do sistema de compactação para reconstruir blocos individuais de imagem.

2. Quantização

De acordo com uma possível definição, quantização é um termo usado para uma
5 função de mapeamento não-reversível de aproximação comumente usada para compactação com perda, na qual existe um conjunto especificado de possíveis valores de saída, e cada membro do conjunto de possíveis valores de saída tem um conjunto associado de valores de entrada que resultam na seleção daquele valor de saída específico. Diversas técnicas de quantização foram desenvolvidas, incluindo quantização escalar ou de vetor, uniforme ou não-uniforme, com ou sem zona morta, e quantização adaptativa ou não-adaptativa.
10

A operação de quantização é essencialmente uma divisão influenciada por um parâmetro de quantização QP que é realizado no codificador. A operação de multiplicação ou quantização inversa é uma multiplicação por QP realizada no decodificador. Esses processos em conjunto introduzem uma perda nos dados originais do coeficiente de transformada, que aparecem como erros ou artefatos de compactação na imagem decodificada. Em um
15 codec simplista, certo valor fixo de QP pode ser aplicado a todos os coeficientes de transformada em um quadro. Embora isso possa ser uma solução aceitável em alguns casos, ele tem várias deficiências: o sistema visual humano não é igualmente sensível a todas as frequências, a todos os locais espaciais dentro de um quadro, ou a todos os canais de luminância e cromaticidade. Utilizando diferentes valores QP para diferentes coeficientes pode
20 prover uma codificação visualmente superior mesmo com o mesmo número ou com um número menor de bits compactados. Similarmente, outras métricas de erro podem ser também adequadamente otimizadas.

O controle de taxa ou a capacidade de um codificador produzir um arquivo compactado de um tamanho desejado não é fácil de realizar com um único QP através do quadro
25 inteiro.

Portanto, é desejável permitir que o codificador varie o QP através da imagem de uma maneira arbitrária. Contudo, isso significa que o valor efetivo de QP usado para cada
30 partição de dados deve ser sinalizado no fluxo de bits. Isso leva a um enorme overhead apenas para transportar a informação de sinalização QP, tornando o mesmo inadequado na prática. O que é desejado é um meio econômico em termos de bits, ainda assim flexível, de sinalizar QP, particularmente para cenários comumente encontrados.

Em resumo, a quantização é o mecanismo principal para a maioria dos codecs de
35 imagem e vídeo controlar a qualidade da imagem compactada e a taxa de compactação. Os métodos de quantização suportados pelos codecs mais populares proporcionam poucos recursos ou pouca flexibilidade, ou incorrem em overhead significativo de bits adicionais. Frequentemente, uma imagem ou um quadro de vídeo é normalmente quantizado unifor-

memente, ou com capacidade limitada de variar a quantização através de locais espaciais. Essa ausência de flexibilidade prejudica a qualidade da compactação, e impede instantâneo controle exato da taxa. Por outro lado, alguns codecs podem proporcionar liberdade quase irrestrita no suporte aos métodos de quantização. A codificação para uso de sinais de quantizadores diferentes considera bits adicionais nos meios codificados, e ele próprio poderia afetar adversamente a eficácia da compactação. Adicionalmente, o processo de construir um decodificador correspondente requer um grande número de etapas de teste, gerado por todas as possíveis combinações dos métodos de quantização, o que pode ser oneroso.

Sumário

10 A Descrição Detalhada a seguir apresenta variações de uma técnica de quantização flexível que provê a capacidade de variar a quantização ao longo de várias dimensões dos dados de meios digitais codificados. Por exemplo, uma implementação representativa da técnica de quantização flexível pode variar a quantização em relação a três dimensões – em relação a: (i) locais espaciais, (ii) sub-bandas de frequência, e (iii) canais de cores. A
15 Descrição Detalhada apresenta adicionalmente formas de eficientemente sinalizar a quantização flexível nos dados de meios digitais codificados. A vantagem dessa abordagem de quantização é que o overhead incorrido pela informação secundária relacionada à quantização é minimizado para os cenários de utilização principal, enquanto permitindo máxima flexibilidade se desejada pelo codificador.

20 Esse Resumo é provido para introduzir uma seleção dos conceitos em uma forma simplificada que é descrita adicionalmente na Descrição Detalhada. Esse sumário não pretende identificar características fundamentais ou características essenciais da matéria reivindicada, nem pretende ser usado como um meio auxiliar na determinação do escopo da matéria reivindicada. Características e vantagens adicionais da invenção se tornarão evidentes a partir da descrição detalhada seguinte das modalidades que prosseguem com referência aos desenhos anexos.

Descrição Resumida dos Desenhos

A Figura 1 é um diagrama de blocos de um codec baseado em transformada por blocos, convencional, na técnica anterior.

30 A Figura 2 é um diagrama de fluxo de um codificador representativo incorporando a codificação de padrão de blocos.

A Figura 3 é um diagrama de fluxo de um decodificador representativo incorporando a codificação de padrão de blocos.

35 A Figura 4 é uma tabela contendo uma definição de pseudocódigos para sinalização de um quantizador DC de acordo com uma técnica de quantização flexível.

A Figura 5 é uma tabela contendo uma definição de pseudocódigo para sinalização de um quantizador passa-baixa de acordo com a técnica de quantização flexível.

A Figura 6 é uma tabela contendo uma definição de pseudocódigo para sinalização de um quantizador passa-alta de acordo com a técnica de quantização flexível.

5 A Figura 7 é uma tabela contendo uma definição de pseudocódigo para sinalização de quantizadores em uma camada de quadro de acordo com a técnica de quantização flexível.

A Figura 8 é uma tabela contendo uma definição de pseudocódigo para sinalização de quantizadores em uma camada de tiles no modo espacial de acordo com a técnica de quantização flexível.

10 A Figura 9 é uma tabela contendo uma definição de pseudocódigo para sinalização de quantizadores de uma sub-banda DC na camada de tiles no modo de frequência de acordo com a técnica de quantização flexível.

A Figura 10 é uma tabela contendo uma definição de pseudocódigo para sinalização de quantizadores de uma sub-banda passa-baixa na camada de tiles no modo de frequência de acordo com a técnica de quantização flexível.

15 A Figura 11 é uma tabela contendo uma definição de pseudocódigo para sinalização dos quantizadores de uma sub-banda passa-alta na camada de tiles no modo de frequência de acordo com a técnica de quantização flexível.

A Figura 12 é uma tabela contendo uma definição de pseudocódigo para sinalização de quantizadores em uma camada de macrobloco no modo espacial de acordo com a técnica de quantização flexível.

20 A Figura 13 é uma tabela contendo uma definição de pseudocódigo para sinalização de quantizadores passa-baixa na camada de macrobloco no modo de frequência de acordo com a técnica de quantização flexível.

A Figura 14 é uma tabela contendo uma definição de pseudocódigo para sinalização de quantizadores passa-alta na camada de macrobloco no modo de frequência de acordo com a técnica de quantização flexível.

A Figura 15 é um diagrama de blocos de um ambiente de computação adequado para implementar um codificador/decodificador de meios com quantização flexível.

Descrição Detalhada

30 A descrição a seguir se refere às técnicas de codificação e decodificação que proporcionam quantização flexível eficiente que pode variar a quantização aplicada ao longo das dimensões espaciais, de frequência e cor (referidos aqui como “quantização flexível”). A descrição a seguir descreve uma implementação exemplar da técnica no contexto de um sistema de compactação de meios digitais ou codec. O sistema de meios digitais codifica os dados de meios digitais em uma forma compactada para transmissão ou armazenamento, e
35 decodifica os dados para reprodução ou outro processamento. Com o propósito de ilustração, esse sistema de compactação exemplar incorporando essa quantização flexível é um

sistema de compactação de imagem ou vídeo. Alternativamente, a técnica também pode ser incorporada nos sistemas de compactação ou codecs para outros dados 2D. A técnica de quantização flexível não requer que o sistema de compactação de meios digitais codifique os dados de meios digitais compactados em um formato de codificação específico.

5 1Codificador/Decodificador

As Figuras 2 e 3 são um diagrama generalizado dos processos empregados em um codificador de dados bidimensional (2D) representativo 200 e decodificador 300. Os diagramas apresentam uma ilustração generalizada ou simplificada de um sistema de compactação incorporando o codificador de dados 2D e decodificador que implementam a codificação de padrão de blocos. Em sistemas de compactação alternativos utilizando a codificação de padrão de blocos, processos adicionais, ou um número menor de processos do que aqueles ilustrados nesse codificador e decodificador representativos podem ser usados para a compactação de dados 2D. Por exemplo, alguns codificadores/decodificadores também podem incluir conversão de cor, formatos de cor, codificação escalonável, codificação sem perda, modos de macrobloco, etc. O sistema de compactação (codificador e decodificador) pode prover compactação de dados 2D sem perda e/ou com perda, dependendo da quantização a qual pode se basear em um parâmetro de quantização variando de sem perda para com perda.

O codificador de dados 2D 200 produz um fluxo de bits compactado 220 que é uma representação mais compacta (para entrada típica) de dados 2D 210 apresentados como entrada ao codificador. Por exemplo, a entrada de dados 2D pode ser uma imagem, um quadro de uma seqüência de vídeo, ou outros dados tendo duas dimensões. O codificador de dados 2D divide um quadro dos dados de entrada em blocos (ilustrado geralmente na Figura 2 como partição 230), os quais na implementação ilustrada são blocos de 4x4 pixels não sobrepostos que formam um padrão regular através do plano do quadro. Esses blocos são agrupados em clusters, denominados macroblocos, os quais são de 16x16 pixels de tamanho nesse codificador representativo. Por sua vez, os macroblocos são agrupados em estruturas regulares denominados tiles. Os tiles também formam um padrão regular sobre a imagem, de tal modo que os tiles em uma fileira horizontal têm altura uniforme, e são alinhados, e os tiles em uma coluna vertical têm largura uniforme e são alinhados. No codificador representativo, os tiles podem ser de qualquer tamanho arbitrário quer seja um múltiplo de 16 na direção horizontal e/ou vertical. Implementações alternativas de codificador podem dividir a imagem em blocos, macroblocos, tiles ou outras unidades de outros tamanhos e estruturas.

Um operador de “sobreposição direta” 240 é aplicado a cada borda entre blocos, após o que cada bloco 4x4 é transformado utilizando uma transformada por blocos 250. Essa transformada por blocos 250 pode ser a transformada 2D reversível, de escala livre des-

crita por Srinivasan, Pedido de Patente dos Estados Unidos 11/015.707 intitulado “Reversible Transform For Lossy And Lossless 2-D Data Compression”, depositado em 17 de dezembro de 2004. O operador de sobreposição 240 pode ser o operador de sobreposição reversível descrito por Tu e outros, Pedido de Patente dos Estados Unidos 11/015.148, intitulado “Reversible Overlap Operator for Efficient Lossless Data Compression”, depositado em 17 de dezembro de 2004; e por Tu e outros, Pedido de Patente dos Estados Unidos 11/035.991 intitulado “Reversible 2-Dimensional Pre-/Post-Filtering For Lapped Biorthogonal Transform”, depositado em 14 de janeiro de 2005. Alternativamente, a transformada discreta do co-seno ou outras transformadas por blocos e os operadores de sobreposição podem ser usados. Subseqüente à transformada, o coeficiente DC 260 de cada bloco de transformada 4x4 é submetido a um encadeamento de processamento similar (disposição lado a lado, sobreposição direta, seguida por transformada de bloco 4x4). Os coeficientes de transformada DC, resultantes e os coeficientes de transformada AC são quantizados 270, codificados por entropia 280 e empacotados 290.

O decodificador realiza o processo inverso. Pelo lado do decodificador, os bits de coeficiente de transformada são extraídos 310 a partir de seus pacotes respectivos, a partir dos quais os coeficientes são eles próprios decodificados 320 e desquantizados 330. Os coeficientes DC 340 são regenerados mediante aplicação de uma transformada inversa, e o plano dos coeficientes DC é “sobreposto inverso” utilizando um operador de suavização adequado aplicado através das bordas de bloco DC. Subseqüentemente, os dados inteiros são regenerados mediante aplicação da transformada inversa 4x4 350 aos coeficientes DC, e os coeficientes AC 342 são decodificados a partir do fluxo de bits. Finalmente, as bordas de bloco nos planos de imagem resultante são filtradas com sobreposição inversa 360. Isso produz uma saída reconstruída de dados 2D.

Na implementação exemplar, o codificador 200 (Figura 2) compacta uma imagem de entrada no fluxo de bits compactado 220 (por exemplo, um arquivo), e o decodificador 300 (Figura 3) reconstrói a entrada original ou uma aproximação da mesma, com base no fato de se é empregada a codificação sem perda ou a codificação com perda. O processo de codificação envolve aplicação de uma transformada sobreposta direta (LT) discutida abaixo, a qual é implementada com pré-/pós-filtração bidimensional reversível também descrita mais completamente abaixo. O processo de decodificação envolve a aplicação da transformada sobreposta inversa (ILT) utilizando a pré-/pós-filtração bidimensional reversível.

A LT e a ILT, ilustradas, são inversos uma da outra, em um sentido exato e, portanto, podem ser referidas coletivamente como transformada sobreposta, reversível. Como uma transformada reversível, o par de LT/ILT pode ser usado para compactação de imagem sem perda.

Os dados de entrada 210 compactados pelo codificador 200/decodificador 300, ilus-

trado, podem ser imagens de vários formatos de cor (por exemplo, formatos de imagem em cores RGB/YUV4:4:4, YUV4:2:2 ou YUV4:2:0). Tipicamente, a imagem de entrada sempre tem um componente de luminância (Y). Se ela é uma imagem RGB/YUV4:4:4, YUV4:2:2 ou YUV4:2:0, a imagem tem também componentes de crominância, tal como um componente U e um componente V. Os planos de cores separados ou os componentes da imagem podem ter diferentes resoluções espaciais. No caso de uma imagem de entrada no formato de cor YUV4:2:0, por exemplo, os componentes U e V têm metade da largura e altura do componente Y.

Conforme discutido acima, o codificador 200 arranja lado a lado a imagem de entrada ou figura em macroblocos. Em uma implementação exemplar, o codificador 200 arranja lado a lado as imagens de entrada em áreas de 16x16 pixels (denominadas "macroblocos") no canal Y (as quais podem ser áreas de 16x16, 16x8 ou 8x8 nos canais U e V dependendo do formato de cor). Cada plano de cor de macrobloco é arranjado lado a lado em regiões ou blocos de 4x4 pixels. Portanto, um macrobloco é composto para vários formatos de cores da seguinte maneira para essa implementação de codificador exemplar:

1. Para uma imagem em tons de cinza, cada macrobloco contém 16 blocos de 4x4 luminância (Y).

2. Para uma imagem em cores de formato YUV4:2:0, cada macrobloco contém 16 blocos Y 4x4, e 4 blocos de crominância 4x4 (U e V).

3. Para uma imagem em cores de formato YUV4:2:2, cada macrobloco contém 16 blocos Y 4x4, e 8 blocos de crominância 4x4 (U e V).

4. Para um RGB ou imagem em cores YUV4:4:4, cada macrobloco contém 16 blocos cada de canais Y, U e V.

Conseqüentemente, após a transformada, um macrobloco nesse codificador 200/decodificador 300 representativo tem três sub-bandas de freqüência: uma sub-banda DC (macrobloco DC), uma sub-banda passa-baixa (macrobloco passa-baixa), e uma sub-banda passa-alta (macrobloco passa-alta). No sistema representativo, as sub-bandas passa-baixa e/ou passa-alta são opcionais no fluxo de bits – essas sub-bandas podem ser completamente descartadas.

Adicionalmente, os dados compactados podem ser adensados no fluxo de bits em um de dois ordenamentos: ordem espacial e ordem de freqüência. Para a ordem espacial, diferentes sub-bandas do mesmo macrobloco dentro de um tile são ordenadas em conjunto, e o fluxo de bits resultante de cada tile é gravado em um pacote. Para a ordem de freqüência, a mesma sub-banda a partir de macroblocos diferentes dentro de um tile são agrupadas em conjunto, e desse modo o fluxo de bits de um tile é gravado em três pacotes: um pacote de tiles DC, um pacote de tiles passa-baixa, e um pacote de tiles passa-alta. Além disso, pode haver outras camadas de dados.

Desse modo, para o sistema representativo, uma imagem é organizada nas seguintes “dimensões”:

Dimensão espacial: Quadro → tile → Macrobloco;

Dimensão de frequência: DC|Passa-baixa|Passa-alta; e

5 Dimensão de canal: Luminância|Crominância_0|Crominância_1 ... (por exemplo, como Y|U|V).

As setas acima denotam uma hierarquia, enquanto que as barras verticais denotam uma partição.

Embora o sistema representativo organize os dados de meios digitais compactados em dimensões espaciais, de frequência e de canal, a abordagem de quantização flexível aqui descrita pode ser aplicada nos sistemas de codificador/decodificador alternativos que organizam seus dados ao longo de um número menor de dimensões, de dimensões adicionais ou outras dimensões. Por exemplo, a abordagem de quantização flexível pode ser aplicada para codificação utilizando um número maior de bandas de frequência, outro formato de canais de cores (por exemplo, YIQ, RGB, etc.), canais de imagem adicionais (por exemplo, para visão estérea ou outros arranjos de múltiplas câmeras).

2Visão Geral da Quantização Flexível

No codificador/decodificador representativo, a operação de quantização é essencialmente uma divisão influenciada por um parâmetro de quantização QP o qual é realizado no codificador. A operação de multiplicação ou quantização inversa é uma multiplicação por QP realizada no decodificador. Contudo, implementações alternativas da quantização flexível aqui descrita podem utilizar outras formas de quantização, incluindo uniforme e não-uniforme, escalar ou de vetor, com ou sem zona morta, etc. Os processos de quantização/quantização inversa em conjunto introduzem uma perda nos dados de coeficiente de transformada original, que se apresentam como erros ou artefatos de compactação na imagem decodificada. Em um codec simplista, certo valor fixo de QP pode ser aplicado a todos os coeficientes de transformada em um quadro. Embora essa possa ser uma solução aceitável em alguns casos, ela apresenta várias deficiências:

O sistema visual humano não é igualmente sensível a todas as frequências, ou a todos os locais espaciais dentro de um quadro, ou a todos os canais de luminância e crominância. Ao utilizar valores QP, diferentes, para diferentes coeficientes, se pode prover uma codificação visualmente superior mesmo com o mesmo número ou com um número menor de bits compactados. Similarmente, outras métricas de erro também podem ser adequadamente otimizadas.

O controle de taxa ou a capacidade de um codificador em produzir um arquivo compactado de um tamanho desejado não é fácil de realizar com um único QP através do quadro inteiro.

Portanto, idealmente deve ser possível permitir que o codificador varie o QP através da imagem de uma forma arbitrária. Contudo, isto significa que o valor efetivo de QP usado para cada partição de dados (macrobloco/tile/canal/sub-banda, etc.) deve ser sinalizado no fluxo de bits. Isso leva a um enorme overhead apenas para transportar a informação de sinalização QP, tornando o mesmo inadequado na prática. O que é desejado é um meio econômico em termos de bit ainda assim flexível de sinalizar QP, particularmente para cenários comumente encontrados.

A técnica de quantização flexível aqui descrita provê a capacidade de variar a quantização ao longo de várias partições ou dimensões dos dados de meios digitais codificados. Por exemplo, uma implementação da técnica de quantização flexível no sistema de codificador 200/decodificador 300 representativo pode variar a quantização através de três dimensões – através de: (i) localizações espaciais, (ii) sub-bandas de freqüência, e (iii) canais de cor. Contudo, a quantização pode ser variável através de um número menor de dimensões ou partições, de dimensões ou partições adicionais ou outras dimensões ou partições dos dados em outras implementações alternativas da técnica de quantização flexível. Essa técnica também inclui formas de eficientemente sinalizar a quantização flexível nos dados de meios codificados. O benefício dessa abordagem de quantização é que o overhead incorrido pela informação secundária relacionada à quantização é minimizado para os cenários de utilização principal, enquanto permitindo flexibilidade máxima se desejado pelo codificador.

A técnica de quantização flexível provê controle de granularidade espacial preciso da quantização. Em uma implementação específica, a quantização flexível permite controle através da quantização aplicada ao quadro, tile, ou descendentemente para o macrobloco. Se o quadro não for quantizado uniformemente, então cada tile pode ser quantizado uniformemente; se um tile não for quantizado uniformemente, então cada macrobloco será quantizado diferentemente.

A quantização flexível permite adicionalmente o controle de quantização ao longo da dimensão de sub-banda de freqüência. Em uma implementação específica, a quantização flexível inclui um modo de sub-banda para especificar uma relação de quantização entre sub-bandas de freqüência. As sub-bandas podem ser quantizadas uniformemente, ou parcialmente uniformemente (sub-banda passa-baixa utilizando quantizador de sub-banda DC, e/ou sub-banda passa-alta utilizando quantizador passa-baixa), ou independentemente.

A quantização flexível também permite controle sobre a quantização aplicada ao longo da dimensão de canal dos dados. Em uma implementação específica, a quantização flexível inclui um modo de canal para especificar uma relação de quantização entre canais de cores. Os canais podem ser quantizados uniformemente, ou parcialmente uniformemente (canais de crominância uniformemente, mas luminância independentemente), ou independentemente.

A quantização flexível descrita aqui também provê técnicas para eficientemente sinalizar na informação secundária dos dados de meios digitais compactados, combinações do controle de quantização acima através de sub-banda de freqüência espacial e canal, que são de significância para os cenários de utilização principal. Além disso, a técnica de quantização flexível provê uma forma de eficientemente definir a escolha do quantizador mediante indexação a partir de um subconjunto definido de quantizadores possíveis nos dados de meios digitais.

3. Quantização flexível na dimensão espacial

Na dimensão espacial, são providas três escolhas por intermédio da técnica de quantização flexível no codificador/decodificador representativo:

- O quadro inteiro pode ser codificado utilizando a mesma regra de quantização.
- Do contrário, um tile inteiro pode ser codificado utilizando a mesma regra de quantização e diferentes tiles dentro do quadro podem utilizar diferentes regras de quantização.
- Do contrário, cada macrobloco dentro de um tile pode ser codificado utilizando a mesma regra de quantização e macroblocos diferentes dentro do tile podem usar diferentes regras de quantização.

Um meio de sinalizar essas possibilidades é como a seguir: Um sinal binário é enviado no fluxo de bits no nível de quadro indicando se a primeira possibilidade é verdadeira. Se não for, um símbolo de comprimento fixo é enviado no fluxo de bits dentro de cada tile indicando o número de regras de quantização usadas para esse tile. Se o tile usa mais do que uma regra de quantização, então um símbolo de comprimento variável é enviado dentro de cada macrobloco dentro do tile correspondente que indica a regra de quantização usada pelo macrobloco. O decodificador interpreta o fluxo de bits de uma maneira consistente com o codificador.

O codificador 200/decodificador 300 representativo utiliza uma variante da sinalização acima. Um sinal binário, representado por um elemento de sintaxe genérico, aqui rotulado como "XXX_FRAME_UNIFORM", é apenas enviado no nível de quadro (onde XXX é um mantedor de posição especificando a sub-banda de freqüência específica ou a dimensão de canal do controle de quantizador). No nível de tile, o número de regras distintas de quantizador é enviado em um elemento de sintaxe no nível de tile (XXX_QUANTIZERS) apenas quando o elemento de sintaxe de nível de quadro (XXX_FRAME_UNIFORM) é falso. Se esse número for igual a 1, isso significa que existe apenas uma regra e, portanto, todos os macroblocos dentro do tile são codificados uniformemente com a mesma regra de quantização (indicando a escolha 2), e caso contrário, ele indica a escolha da terceira possibilidade.

4. Quantização flexível através de faixas de freqüência:

Para quantização flexível através de faixas de freqüência, a sintaxe de fluxo de bits do codificador 200/decodificador 300 representativo define dois comutadores:

- O macrobloco de passa-baixa utiliza a mesma regra de quantização que o macrobloco DC na mesma localização espacial. Isso corresponde ao elemento de sintaxe USE_DC_QUANTIZER.

- O macrobloco passa-alta utiliza a mesma regra de quantização que o macrobloco passa-baixa na mesma localização espacial. Isso corresponde ao elemento de sintaxe USE_LP_QUANTIZER.

Esses comutadores são habilitados na camada de quadro quando o quadro inteiro utiliza a mesma regra de quantização, ou ao contrário, na camada de tile. Esses comutadores não são habilitados na camada de macroblocos. Todos os macroblocos dentro de um tile, portanto, obedecem às mesmas regras através de faixas de frequência. Um símbolo binário é enviado para cada um dos comutadores na camada apropriada (quadro ou tile).

5.Quantização flexível através de canais de imagem:

Para quantização flexível através dos canais, a sintaxe de fluxo de bits do codificador 200/decodificador 300 representativo permite três opções:

- Todos os canais – luminância e crominância têm a mesma regra de quantização. Isso é indicado pelo elemento de sintaxe genérico XXX_CH_MODE == CH_UNIFORM.

- A luminância segue uma regra de quantização e todos os canais de crominância seguem uma regra de quantização diferente, indicada por XXX_CH_MODE == CH_MIXED.

- Todos os canais são livres para escolher diferentes regras de quantização, indicadas por XXX_CH_MODE == CH_INDEPENDENT.

6.Quantização flexível combinatorial:

O codificador 200/decodificador 300 representativo utiliza uma sintaxe de fluxo de bits definida nas tabelas de código mostrada nas Figuras 4-14 que podem eficientemente codificar a opção específica dentre as opções de quantização flexível através das dimensões discutidas acima. Com várias opções de quantização, disponíveis através de cada uma de sub-banda de frequência espacial e dimensões de canal, o número de permutações das opções disponíveis de quantização é grande. Aumentando a complexidade da quantização flexível através das três dimensões está o fato de que o fluxo de bits do codificador 200/decodificador 300 representativo pode ser disposto em ordenamento de frequência ou espacial. Contudo, isso não muda as opções de quantização disponíveis, e apenas afeta a serialização dos sinais. A sintaxe definida nas Figuras 4-14 provê uma codificação eficiente das regras de quantização flexíveis combinatorias.

Algumas características destacadas das regras de quantização combinatorias como definidas na sintaxe do codificador/decodificador representativo são conforme a seguir.

- A quantização DC não pode variar em uma base de macrobloco. Isso permite a codificação diferencial de valores DC quantizados sem se ter que realizar uma operação de escalonamento inverso. A codificação da banda DC de um tile de imagem com um quanti-

zador relativamente pequeno mesmo quando as bandas AC (passa baixa e passa alta) são codificadas com quantização variável não afeta consideravelmente a taxa de bits.

Em uma extremidade da escala, todos os coeficientes de transformada dentro de um quadro utilizam o mesmo parâmetro de quantização. Na outra extremidade da escala, regras de quantização de passa-baixa e de passa-alta para todos os canais podem variar independentemente para cada macrobloco do tile /quadro. A única restrição é que o número de regras distintas de quantizador passa-baixa e passa-alta (cobrindo todos os canais) seja individualmente limitado a 16. Cada regra pode especificar valores independentes do parâmetro de quantização para cada canal.

Entre esses extremos, várias combinações são permitidas conforme especificado pelas tabelas de sintaxe mostradas nas Figuras 4-14.

7. Indexação de parâmetros de quantização:

O parâmetro de quantização específico (QP) no codificador/decodificador representativo se baseia em uma escala harmônica. Um valor de 8 bits de um índice de parâmetro de quantização (QPI) corresponde a um valor de QP o qual pode ser relativamente grande. Um segundo nível de indexação é realizado de modo que os QPIs variando através de macroblocos podem ser codificados de uma maneira eficiente.

Mais especificamente, o codificador 200 pode definir um conjunto no fluxo de bits contendo entre 1 e 16 “vetores” QPI. Cada vetor QPI é composto de um ou mais valores QPI, com base nos quais XXX_CHANNEL_MODE é escolhido. Tais conjuntos são definidos para sub-bandas passa-baixa e passa-alta, DC, com base no comutador de faixa de frequência. Além disso, o conjunto DC tem apenas um vetor QPI uma vez que apenas um quantizador DC é permissível em um canal-tile. A codificação desses conjuntos é definida nas tabelas mostradas nas Figuras 4-6.

Conforme mostrado nas tabelas das Figuras 7-11, a sinalização dos conjuntos de vetores QPI de sub-bandas de frequência passa-alta e passa-baixa, DC, ocorre conforme a seguir. Com base em outros modos de codificação, a cardinalidade de cada conjunto (isto é, o número de vetores QPI no conjunto) é indicada para sub-bandas passa-baixa e passa-alta no início do tile ou quadro correspondente. A cardinalidade do conjunto DC é 1. Nas tabelas de pseudocódigos, o elemento de sintaxe, denotando cardinalidade, é rotulado como XXX_QUANTIZERS”.(Na prática, XXX_QUANTIZERS-1 é enviado no fluxo de bits). Os elementos de sintaxe rotulados “XXX_QUANTIZER” nas tabelas denotam a codificação dos conjuntos QPI, que é definida nas tabelas mostradas nas Figuras 4-6.

No nível de macrobloco, é suficiente enviar apenas o índice QI do vetor QPI desejado a partir de dentro do conjunto QPI. As tabelas nas Figuras 12-14 definem a sintaxe do envio de QI em uma base de macrobloco. O elemento de sintaxe correspondendo ao QI é rotulado, “XXX_QUANTIZER_INDEX”. Um código de comprimento variável é usado para

5 sinalizar QI. Primeiro lugar, um símbolo de 1 bit é enviado indicando se QI é zero ou não. Se não for zero, então um código de comprimento fixo sendo determinado por $\text{ceil}(\log_2(\text{XXX_QUANTIZERS}-1))$ é enviado indicando o QI específico diferente de zero. Isso permite uma codificação eficiente de uma regra de quantização “padrão” (QI = 0) com pouco quanto 1 bit por macrobloco. Quando XXX_QUANTIZERS é 1, XXX_QUANTIZER_INDEX é unicamente zero e, portanto, QI não precisa ser sinalizado.

8. Extensões

10 A descrição acima da quantização flexível é específica para sua implementação em um codificador e decodificador representativo, e sintaxe. Contudo, os princípios dessa técnica podem ser estendidos para outros sistemas de compactação de meios digitais assim como formatos. Por exemplo, o codificador/decodificador representativo tem apenas três sub-bandas de frequência (DC, passa-baixa, e passa-alta). Mas, mais geralmente, implementações alternativas da quantização flexível podem ser estendidas de uma maneira direta para uma grande quantidade de sub-bandas de frequência. Similarmente, implementações alter-
15 nativas de quantização flexível podem variar o quantizador em granularidade espacial mais exata, tal como mediante envio de informação de índice de quantização (QI) no nível de sub-macrobloco (tal como bloco). Muitas extensões para os princípios subjacentes da técnica de quantização flexível são possíveis dentro da mesma estrutura.

9. Ambiente de Computação

20 As técnicas de processamento, descritas acima, para quantização flexível, podem ser realizadas em qualquer um de uma variedade de sistemas de codificação e/ou decodificação de meios digitais, incluindo, entre outros exemplos, computadores (de diversos fatores de forma, incluindo: servidor, de mesa, laptop, de mão, etc.); gravadores e aparelhos de reprodução de meios digitais; dispositivos de captura de imagem e vídeo (tais como câmeras, scanners, etc.); equipamento de comunicação (tal como telefones, fones móveis, equipamentos de videoconferência, etc.); display, dispositivos de impressão ou outros dispositivos de apresentação; e etc. As técnicas de quantização flexíveis podem ser implementadas em conjunto de circuitos de hardware, em firmware controlando hardware de processamento de meios digitais, assim como em software de comunicação executando dentro de um com-
25 putador ou outro ambiente de computação, tal como mostrado na Figura 15.

A Figura 15 ilustra um exemplo generalizado de um ambiente de computação adequado (1500) no qual as modalidades descritas podem ser implementadas. O ambiente de computação (1500) não pretende sugerir qualquer limitação em relação ao escopo de uso ou funcionalidade da invenção, uma vez que a presente invenção pode ser implementada em diversos ambientes de computação de uso geral ou de uso especial.
35

Com referência à Figura 15, o ambiente de computação (1500) inclui ao menos uma unidade de processamento (1510) e memória (1520). Na Figura 15, essa configuração

mais básica (1530) é incluída dentro de uma linha tracejada. A linha de processamento (1510) executa instruções executáveis por computador e pode ser um processador real ou virtual. Em um sistema de múltiplo processamento, as unidades de processamento múltiplo executam instruções executáveis por computador para aumentar a capacidade de processamento. A memória (1520) pode ser memória volátil (por exemplo, registradores, cache, RAM), memória não-volátil (por exemplo, ROM, EEPROM, memória flash, etc.), ou alguma combinação das duas. A memória (1520) armazena software (1580) implementando a codificação/decodificação de meios digitais descritas com técnicas de quantização flexível.

Um ambiente de computação pode ter características adicionais. Por exemplo, o ambiente de computação (1500) inclui armazenamento (1540), um ou mais dispositivos de entrada (1550), um ou mais dispositivos de saída (1560), e uma ou mais conexões de comunicação (1570). Um mecanismo de interconexão (não mostrado) tal como um barramento, controlador, ou rede, interconecta os componentes do ambiente de computação (1500). Tipicamente, o software de sistema operacional (não mostrado) provê um ambiente de operação para outro software executando no ambiente de computação (1500), e coordena as atividades dos componentes do ambiente de computação (1500).

O armazenamento (1540) pode ser removível ou não-removível e inclui discos magnéticos, fitas magnéticas ou cassetes, CD-ROMs, CD-RWs, DVDs, ou qualquer outro meio que possa ser usado para armazenar informação e que possa ser acessado dentro do ambiente de computação (1500). O armazenamento (1540) armazena instruções para o software (1580) implementando a codificação/decodificação de meios digitais descritos com técnicas de quantização flexível.

O dispositivo(s) de entrada (1550) pode ser um dispositivo de entrada sensível ao toque tal como um teclado, mouse, caneta, ou trackball, um dispositivo de entrada de voz, um dispositivo de varredura, ou outro dispositivo que proporcione entrada para o ambiente de computação (1500). Para áudio, o dispositivo(s) de entrada (1550) pode ser uma placa de som ou um dispositivo similar que aceita entrada de áudio na forma analógica ou na forma digital a partir de um microfone ou arranjo de microfones, ou uma leitora de CD-ROM que provê amostras de áudio ao ambiente de computação. O dispositivo(s) de saída (1560) pode ser um display, impressora, alto-falante, gravador de CD, ou outro dispositivo que proporcione saída a partir do ambiente de computação (1500).

A conexão(ões) de comunicação (1570) permite a comunicação através de um meio de comunicação para outra entidade de computação. O meio de comunicação transporta informação tal como instruções executáveis por computador, informação de áudio ou vídeo compactado, ou outros dados em um sinal modulado de dados. Um sinal modulado de dados é um sinal que tem uma ou mais de suas características ajustadas ou mudadas de tal modo a codificar informação no sinal. Como exemplo, e não como limitação, os meios de

comunicação incluem técnicas sem fio ou cabeadas implementadas dentro de uma portadora elétrica, ótica, de RF, de infravermelho, acústica ou outra portadora.

5 A codificação/decodificação de meios digitais, aqui descrita, com técnicas de quantização flexível pode ser descrita no contexto geral de meios legíveis por computador. Meios legíveis por computador são quaisquer meios disponíveis que podem ser acessados dentro de um ambiente de computação. Como exemplo, e não como limitação, com o ambiente de computação (1500), meios legíveis por computador incluem memória (1520), armazenador (1540), meios de comunicação, e combinações de quaisquer dos mencionados acima.

10 A codificação/decodificação de meios digitais descritos aqui com técnicas de quantização flexível pode ser descrita no contexto geral de instruções executáveis por computador, tais como aquelas incluídas em módulos de programa, sendo executadas em um ambiente de computação em um processador real ou virtual alvo. Geralmente, os módulos de programa incluem rotinas, programas, bibliotecas, objetos, classes, componentes, estruturas de dados, etc., que realizam tarefas específicas ou implementam tipos de dados abstratos
15 específicos. A funcionalidade dos módulos de programa pode ser combinada ou dividida entre módulos de programa conforme desejado em várias modalidades. As instruções executáveis por computador para módulos de programa podem ser executadas em um ambiente de computação, local ou distribuída.

20 Com a finalidade de apresentação, a descrição detalhada utiliza termos como “determinar”, “gerar”, “ajustar”, e “empregar” para descrever operações de computador em um ambiente de computação. Esses termos são abstrações de alto nível para operações realizadas por um computador, e não devem ser confundidos com as ações realizadas por um ser humano. As operações de computador efetivas correspondendo a esses termos variam dependendo da implementação.

25 Em decorrência das muitas modalidades possíveis as quais os princípios de nossa invenção podem ser aplicados, reivindicamos como nossa invenção todas as tais modalidades que possam ficar compreendidas no escopo e espírito das reivindicações a seguir e de seus equivalentes.

REIVINDICAÇÕES

1. Método de codificar/decodificar meios digitais, **CHARACTERIZADO** por compreender:

particionar os meios digitais ao longo de mais do que uma dimensão;

5 variar a quantização aplicada aos blocos dos meios digitais através de mais do que uma dimensão particionada dos meios digitais; e

codificar/decodificar os blocos quantizados dos meios digitais.

2. Método, de acordo com a reivindicação 1, **CHARACTERIZADO** pelo fato de que a quantização variável inclui quantização variável através de uma dimensão espacial.

10 3. Método, de acordo com a reivindicação 1, **CHARACTERIZADO** pelo fato de que a quantização variável inclui quantização variável através de uma dimensão de frequência.

4. Método, de acordo com a reivindicação 1, **CHARACTERIZADO** pelo fato de que a quantização variável inclui quantização variável através dos canais.

15 5. Método, de acordo com a reivindicação 1, **CHARACTERIZADO** pelo fato de que os canais são canais de cores.

6. Método, de acordo com a reivindicação 1, **CHARACTERIZADO** pelo fato de que a quantização variável inclui quantização variável através de mais do que uma entre sub-banda de frequência espacial e dimensões de canal.

20 7. Método, de acordo com a reivindicação 1, **CHARACTERIZADO** pelo fato de que a quantização variável compreende:

para uma dimensão particionada,

selecionar se quantiza os meios digitais em uma primeira partição da dimensão particionada uniformemente;

25 se selecionou não quantizar a primeira partição uniformemente, selecionar se quantiza os meios digitais em uma segunda partição da dimensão particionada uniformemente; e

se selecionou não quantizar a segunda partição uniformemente, quantizar independentemente os meios digitais através da dimensão.

30 8. Método, de acordo com a reivindicação 7, **CHARACTERIZADO** pelo fato de que a primeira partição é um quadro, a segunda partição é um tile, e a quantização independentemente compreende quantizar independentemente os blocos dos meios digitais dentro de macroblocos com diferentes quantizadores.

35 9. Método, de acordo com a reivindicação 7, **CHARACTERIZADO** pelo fato de que a primeira partição é uma sub-banda de frequência DC, a segunda partição é uma sub-banda de frequência passa-baixa, e a quantização independentemente compreende quantizar independentemente os blocos dos meios digitais dentro da sub-banda de frequência passa-baixa ou uma sub-banda de frequência passa-alta com quantizadores diferentes.

10. Método, de acordo com a reivindicação 7, **CHARACTERIZADO** pelo fato de que

a primeira partição é um canal de crominância, a segunda partição é um canal de luminância, e a quantização independentemente compreende quantizar independentemente os blocos dos meios digitais dentro dos canais com diferentes quantizadores.

5 11. Método, de acordo com a reivindicação 7, **CARACTERIZADO** por compreender ainda ao quantizar independentemente os meios digitais através da dimensão, especificar um conjunto de quantizadores diferentes disponíveis a serem empregados através da dimensão, e indexar uma opção do quantizador para aplicar a um bloco dos meios digitais a partir do conjunto.

10 12. Dispositivo de processamento de meios digitais **CARACTERIZADO** por compreender:

uma memória para armazenar dados de meios digitais; e

15 um processador de meios digitais programado para codificar e/ou decodificar os dados de meios digitais mediante aplicação de operações diretas ou inversas de aplicação de uma transformada aos blocos dos dados de meios digitais, e aplicar quantizadores aos blocos de dados de meios digitais, em que os quantizadores são seletivamente variáveis através de mais do que uma dimensão dos dados de meios digitais.

13. Dispositivo, de acordo com a reivindicação 12, **CARACTERIZADO** pelo fato de que as dimensões incluem ao menos duas de espacial, de frequência e de canal de cores.

20 14. Ao menos uma portadora de armazenamento de programa tendo código de programa armazenado na mesma para fazer com que um dispositivo de processamento de meios digitais execute o método de processar dados de meios digitais de acordo com um codec, o método **CARACTERIZADO** por compreender:

25 selecionar combinações dos modos de quantização a serem aplicados aos dados de meios digitais variando através de mais do que uma das dimensões espaciais, de sub-banda de frequência e de canal de cor;

aplicar uma transformada de bloco aos dados de meios digitais;

quantizar os dados de meios digitais de acordo com as combinações selecionadas dos modos de quantização; e

30 codificar os dados de meios digitais quantizados em um fluxo de dados compactados; e

sinalizar as combinações selecionadas de modos de quantização no fluxo de dados compactados.

35 15. Ao menos uma portadora de armazenamento de programa, de acordo com a reivindicação 14, **CARACTERIZADA** pelo fato de que os modos de quantização, disponíveis, para quantização variável através da dimensão espacial compreendem um modo de quantização uniforme de um quadro dos dados de meios digitais, um modo para quantização uniforme de um tile dos dados de meios digitais, e um modo para quantizar independen-

temente cada macrobloco dos dados de meios digitais.

16. Ao menos uma portadora de armazenamento de programa, de acordo com a reivindicação 14, **CHARACTERIZADA** pelo fato de que os modos de quantização disponíveis para quantização variável através da dimensão de sub-banda de frequência compreendem
5 um modo para quantização uniforme através das sub-bandas de frequência dos dados de meios digitais, um modo para parcialmente quantizar uniformemente através das sub-bandas de frequência dos dados de meios digitais, e um modo para independentemente quantizar as sub-bandas de frequência dos dados de meios digitais.

17. Ao menos uma portadora de armazenamento de programa, de acordo com a
10 reivindicação 16, **CHARACTERIZADA** pelo fato de que o modo para uniformemente quantizar através das sub-bandas de frequência compreende uma sub-banda de frequência utilizando um quantizador de outra sub-banda de frequência.

18. Ao menos uma portadora de armazenamento de programa, de acordo com a
15 reivindicação 14, **CHARACTERIZADA** pelo fato de que os modos de quantização disponíveis para quantização variável através da dimensão de canal de cor compreendem um modo para quantização uniforme através dos canais de cor dos dados de meios digitais, um modo para parcialmente quantizar uniformemente através dos canais de cor dos dados de meios digitais, e um modo para quantizar independentemente os canais de cor dos dados de meios digitais.

20 19. Ao menos uma portadora de armazenamento de programa, de acordo com a reivindicação 14, **CHARACTERIZADA** pelo fato de que o método compreende ainda, quando um modo para uma partição dos dados de meios digitais em uma dimensão é variável, definir um conjunto de quantizadores disponíveis e especificar um quantizador atual mediante indexação a partir do conjunto.

25 20. Ao menos uma portadora de armazenamento de programa tendo código de programa armazenado na mesma para fazer com que um dispositivo de processamento de meios digitais realize um método de processar dados de meios digitais de acordo com um codec, o método **CHARACTERIZADO** por compreender a decodificação de um fluxo de dados compactados codificado de acordo com o método da reivindicação 14.

Figura 1
Técnica Anterior

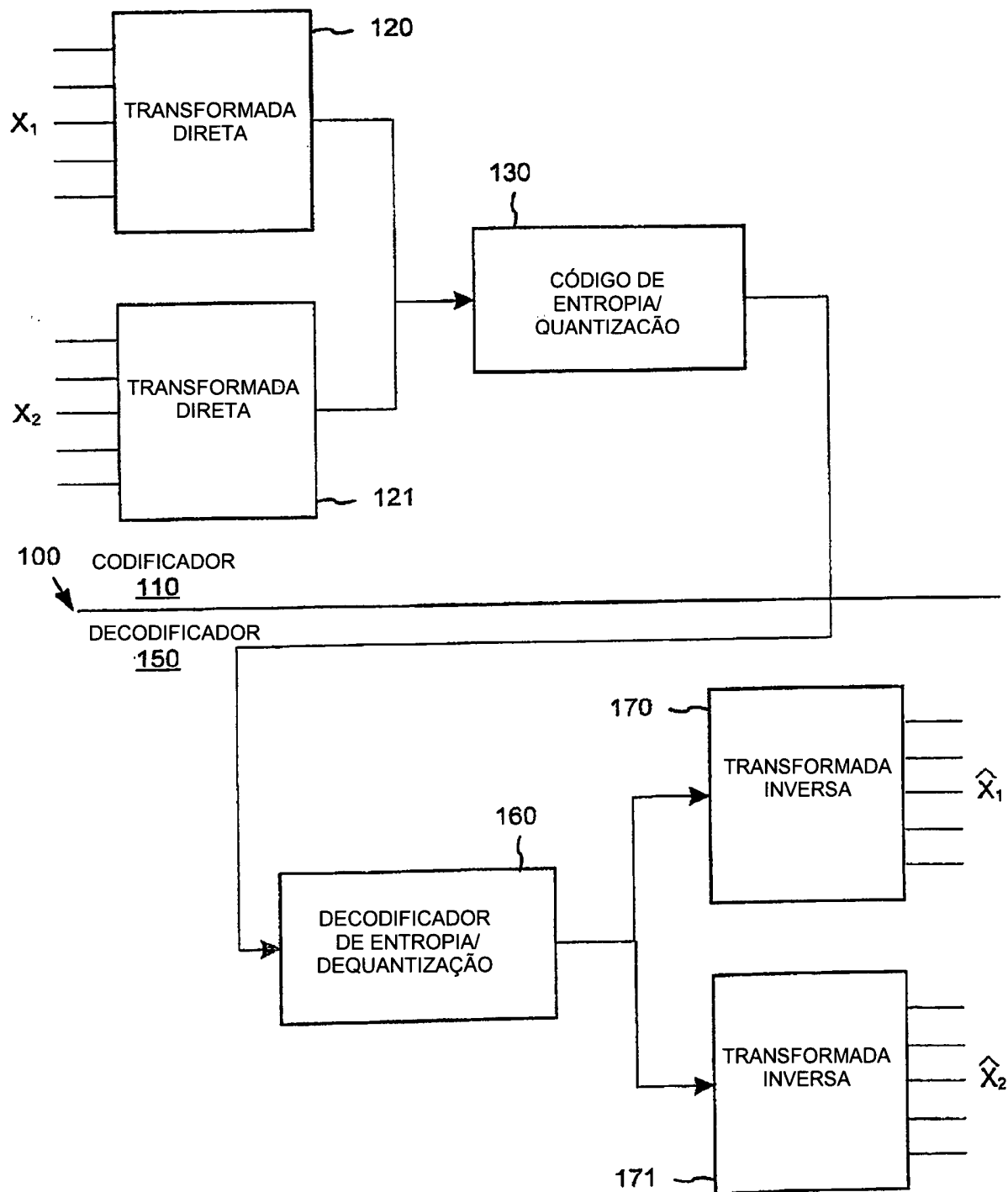


Figura 2

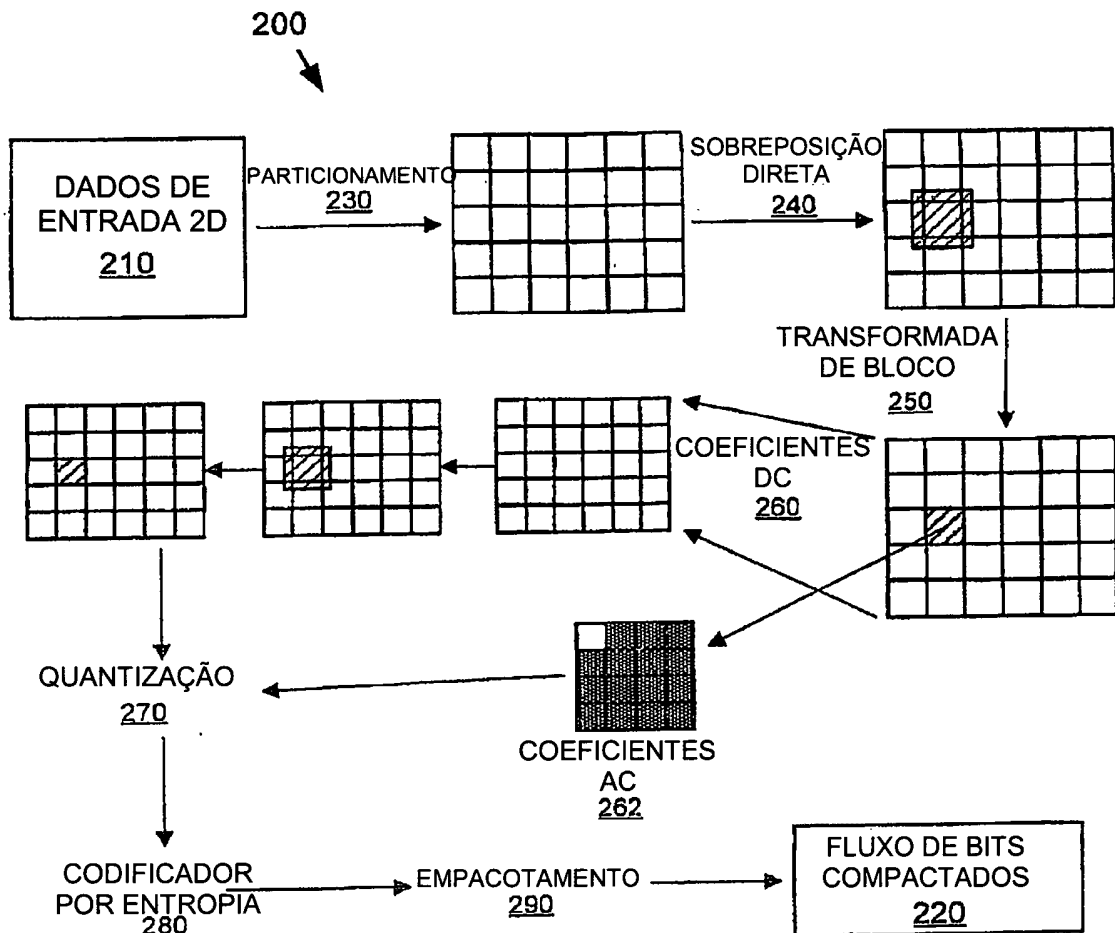


Figura 3

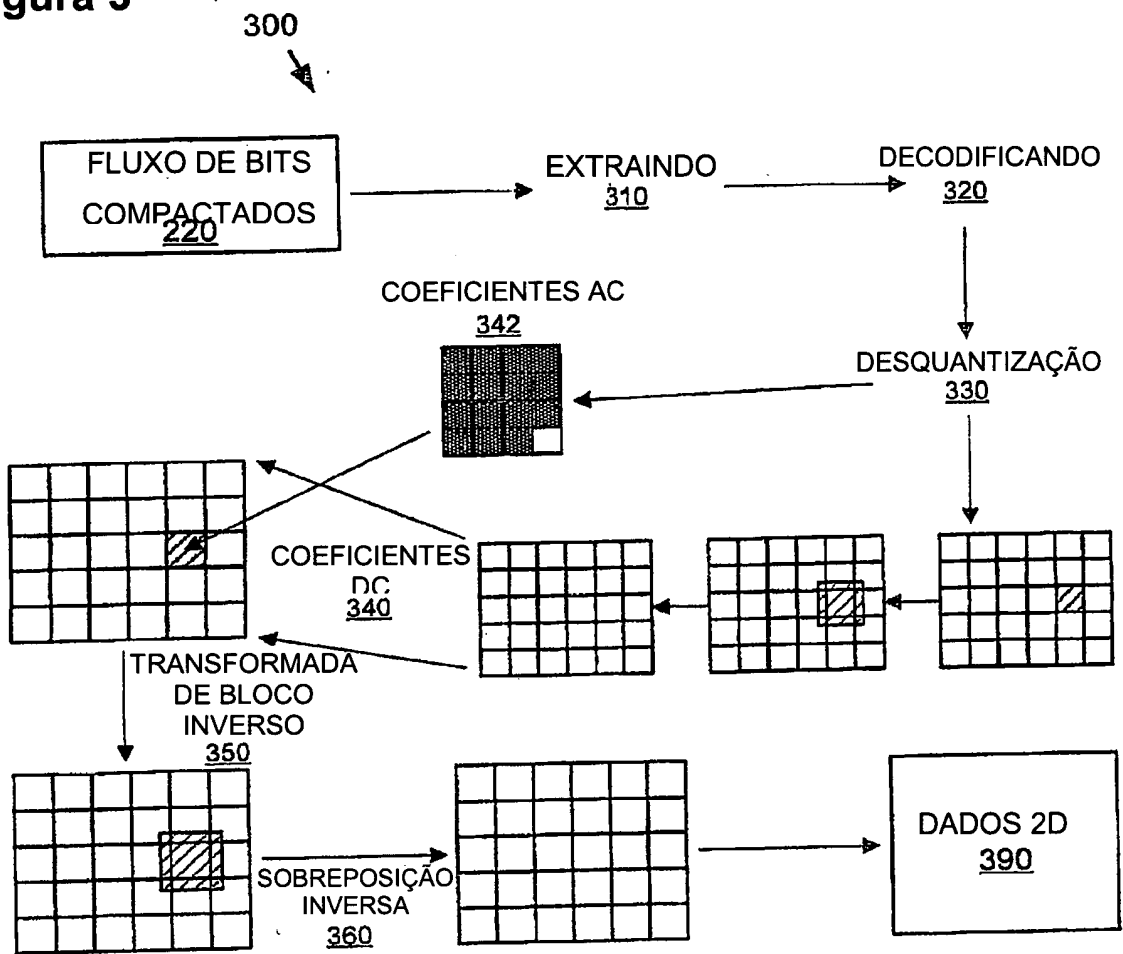


Figura 4

400

DC_QUANTIZER () {	Num bits
se (CHANNELS == 1) DC_CH_MODE = CH_UNIFORM	
Do contrário DC_CH_MODE	2
se (DC_CH_MODE == CH_UNIFORM) DC_QUANT	8
Do contrário se (DC_CH_MODE == CH_MIXED) { DC_QUANT_Y	8
DC_QUANT_UV	8
}	
Do contrário se (DC_CH_MODE == CH_INDEPENDENT) { for (i = 0; i < CHANNELS; i++) DC_QUANT_CHANNEL_i	8
}	
}	

Figura 5

500

LP_QUANTIZER () {	Num bits
for (q = 0; q < LP_QUANTIZERS; q++) { se (CHANNELS == 1) LP_CHMODE[q] = CH_UNIFORM	
Do contrário LP_CH_MODE[q]	2
se (LP_CH_MODE[q] == CH_UNIFORM) LP_QUANT[q]	8
Do contrário se (LP_CH_MODE[q] == CH_MIXED) { LP_QUANT_Y[q]	8
LP_QUANT_UV[q]	8
}	
Do contrário se (LP_CH_MODE[q] == CH_INDEPENDENT) for (i = 0; i < CHANNELS; i++) LP_QUANT_CHANNEL_[q]	8
}	
}	

Figura 6

600	HP_QUANTIZER () {	Num bits
	for (q = 0; q < HP_QUANTIZERS; q++) {	
	se (CHANNELS == 1)	
	HP_CH_MODE[q] = CH_UNIFORM	
	Do contrário	
	HP_CH_MODE[q]	2
	se (HP_CH_MODE[q] == CH_UNIFORM)	
	HP_QUANT[q]	8
	do contrário se (LP_CH_MODE[q] ==	
	CH_MIXED) {	
	HP_QUANT_Y[q]	8
	HP_QUANT_UV[q]	8
	}	
	do contrário se (HP_CH_MODE[q] ==	
	CH_INDEPENDENT)	
	for (i = 0; i < CHANNELS; i++)	
	HP_QUANT_CHANNEL_i[q]	8
	}	
	}	

Figura 7

700	FRAME () {	Num bits
	...	lots
	DC_FRAME_UNIFORM	1
	se (DC_FRAME_UNIFORM)	
	DC_QUANTIZER	variável
	se ('passa-baixa não-saltada'){	
	USE_DC_QUANTIZER	1
	se (USE_DC_QUANTIZER == FALSE){	
	LP_FRAME_UNIFORM	1
	se (LP_FRAME_UNIFORM){	
	LP_QUANTIZERS = 1	
	LP_QUANTIZER()	variável
	}	
	}	
	se ('passa-alta não-saltada'){	
	USE_LP_QUANTIZER	1
	se (USE_LP_QUANTIZER == FALSE){	
	HP_FRAME_UNIFORM	1
	se (HP_FRAME_UNIFORM){	
	HP_QUANTIZERS = 1	
	HP_QUANTIZER()	variável
	}	
	}	
	}	
	}	

Figura 8

	Num bits
800	
SPATIAL_TILE () {	
...	grande quantidade
se (DC_FRAME_UNIFORM == FALSE)	
DC_QUANTIZER	variável
se ('passa-baixa não-saltada' && LP_FRAME_UNIFORM == FALSE) {	
USE_DC_QUANTIZER	1
se (USE_DC_QUANTIZER == FALSE)	
LP_QUANTIZERS	4
LP_QUANTIZER	
}	variável
se ('passa-alta não-saltada' && HP_FRAME_UNIFORM)	
USE_LP_QUANTIZER	1
se (USE_LP_QUANTIZER == FALSE)	
HP_QUANTIZERS	4
HP_QUANTIZER	variável
}	
... ..	
}	

Figura 9

	Num bits
900	
DC_TILE () {	
...	grande quantidade
se (DC_FRAME_UNIFORM == FALSE) {	
DC_QUANTIZER	variável
}	
...	
}	grande quantidade

Figura 10

	Num bits
1000	
LP_TILE () {	
...	grande quantidade
se (LP_FRAME_UNIFORM == FALSE) {	
USE_DC_QUANTIZER	1
se (USE_DC_QUANTIZER == FALSE)	
LP_QUANTIZERS	4
LP_QUANTIZER	
}	
...	
}	grande quantidade

Figura 11

	Num bits
HP_TILE () {	
...	grande quantidade
se (HP_FRAME_UNIFORM == FALSE) {	
USE_LP_QUANTIZER	1
se (USE_LP_QUANTIZER == FALSE)	
HP_QUANTIZERS	4
HP_QUANTIZER	variável
}	
...	grande quantidade
}	

Figura 12

	Num bits
SPATIAL_MB () {	
se ('passa-baixa não-saltada' && LP_QUANTIZERS > 1 && USE_DC_QUANTIZER == FALSE)	
LP_QUANTIZER_INDEX	variável
se ('passa-alta não-saltada' && HP_QUANTIZERS > 1 && USE_HP_QUANTIZER == FALSE)	
HP_QUANTIZER_INDEX	variável
...	grande quantidade
}	

Figura 13

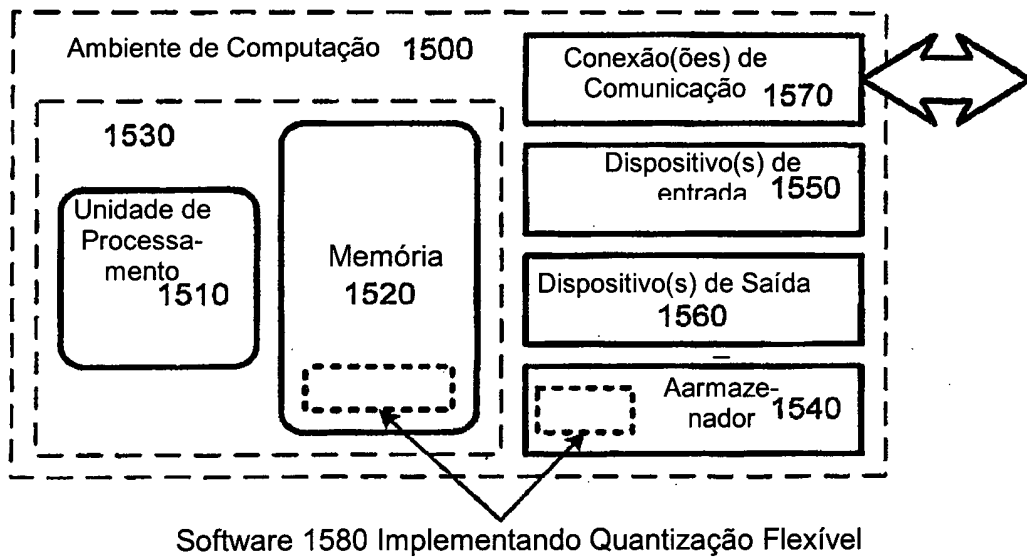
	Num bits
LP_MB () {	
...	grande quantidade
se (LP_QUANTIZERS > 1 && USE_DC_QUANTIZER == FALSE)	
LP_QUANTIZER_INDEX	variável
...	grande quantidade
}	

Figura 14

HP_MB () {	Num bits
...	grande quantidade
se (HP_QUANTIZERS > 1 && USE_LP_QUANTIZER == FALSE)	
HP_QUANTIZER_INDEX	variável
...	grande quantidade
}	

1400

Figura 15



RESUMO

"QUANTIZAÇÃO FLEXÍVEL"

Um codificador/decodificador de meios digitais utiliza uma técnica de quantização flexível que provê a capacidade de variar a quantização ao longo de várias dimensões dos dados de meios digitais codificados, incluindo espacial, subfaixas de freqüência e canais de cor. O codec utiliza um esquema de sinalização para sinalizar várias permutações das combinações de quantização flexível eficientemente para cenários de utilização principal. Quando uma escolha do quantizador está disponível, o codec codifica eficientemente o quantizador atual mediante definição de um subconjunto de quantizadores e indexa o quantizador atual a partir do conjunto.