



REPÚBLICA FEDERATIVA DO BRASIL
MINISTÉRIO DA ECONOMIA
INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL

CARTA PATENTE Nº BR 102013020622-9

O INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL concede a presente PATENTE DE INVENÇÃO, que outorga ao seu titular a propriedade da invenção caracterizada neste título, em todo o território nacional, garantindo os direitos dela decorrentes, previstos na legislação em vigor.

(21) Número do Depósito: BR 102013020622-9

(22) Data do Depósito: 13/08/2013

(43) Data da Publicação Nacional: 27/02/2018

(51) Classificação Internacional: H04N 19/122; H04N 19/18.

(52) Classificação CPC: H04N 19/122; H04N 19/18.

(30) Prioridade Unionista: GB 1214414.3 de 13/08/2012; US 13/584,005 de 13/08/2012; EP 13002521.6 de 14/05/2013.

(54) Título: CODIFICADOR E MÉTODO

(73) Titular: GURULOGIC MICROSYSTEMS OY, Sociedade Finlandesa. Endereço: Linnankatu 34, 20100 Turku, FINLÂNDIA(FI)

(72) Inventor: OSSI KALEVO; TUOMAS KARKKAINEN.

Prazo de Validade: 20 (vinte) anos contados a partir de 13/08/2013, observadas as condições legais

Expedida em: 06/12/2022

Assinado digitalmente por:

Liane Elizabeth Caldeira Lage

Diretora de Patentes, Programas de Computador e Topografias de Circuitos Integrados

Relatório Descritivo da Patente de Invenção para "**CODIFICADOR E MÉTODO**".

CAMPO TÉCNICO

A presente divulgação refere-se a codificadores para receber dados de entrada e codificar os dados de entrada para gerar dados de saída codificados correspondentes. Além do mais, a presente divulgação também diz respeito a métodos de codificar dados de entrada para gerar dados de saída codificados correspondentes. Além disso, a presente divulgação também se refere a produtos de software gravados em meio de armazenamento de dados legíveis por máquina não transitórios, em que os produtos de software são executáveis em hardware de computação para implementar os métodos referidos acima.

ANTECEDENTES

O conteúdo dos dados é armazenado e comunicado em uma extensão crescente pela população humana contemporânea, por exemplo, conteúdo de multimídia através da Internet e meios de comunicação sem fio. Tal conteúdo de multimídia frequentemente inclui, por exemplo, imagens, vídeo e áudio, mas não está limitado a esses. O conteúdo de dados é armazenado e comunicado entre dispositivos, aplicações de software, sistemas de mídia e serviços de dados. Durante esse armazenamento e comunicação, as situações surgem onde imagens e vídeos são capturados, escaneados, transmitidos, compartilhados, monitorados e impressos. No entanto, tais imagens e vídeos são exigentes com respeito à capacidade de memória de dados e à banda larga do sistema de comunicação utilizado. Quando a banda larga do sistema de comunicação é limitada, tais imagens e vídeos tomam um tempo significativo para se comunicarem. Para endereçar tais requisitos de armazenamento, é prática comum utilizar métodos de comunicação de imagem e de vídeo que também proporcionam um grau de compressão de dados. Alguns padrões de codificação comuns para imagens e vídeos são providos na tabela 1.

TABELA 1: PADRONIZAÇÃO DE CODIFICAÇÃO CONTEMPORÂNEA

JPEG	MPEG-1	H.261	WebP	Lucid
JPEG2000	MPEG-2	H.263	WebM	GIF

JPEG XR	MPEG-4	H.264		PNG
	MPEG-4 AVC	HEVC		TIFF
	MPEG-4 MVC			BMP
	MP3			VC-1
				Theora
				AAC
				FLAC
				Ogg Vorbis
				Speex

Arquivos de imagem e de áudio tornam-se maiores na medida em que a qualidade da imagem é progressivamente melhorada, por exemplo, pela adição de padrões de alta definição (HD) e de faixa dinâmica grande (HDR). Imagens tridimensionais (3-D), vídeos e áudio estão ganhando popularidade crescente que demanda codificação correspondentemente mais eficaz e métodos de decodificação em codificadores e decodificadores, a saber, "codecs", para lidar com as quantidades de dados aumentadas associadas a serem armazenados. No entanto, é altamente desejável que os métodos de codificação que proporcionam um grau de compressão de dados devam ser substancialmente sem perda em relação ao conteúdo da informação quando gerando os dados comprimidos.

Codificadores/decodificadores convencionais são descritos em pedidos de patentes publicadas e patentes concedidas, por exemplo, como providos na tabela 2.

15 **TABELA 2: LITERATURA ANTERIOR DESCREVENDO CODIFICADORES/DECODIFICADORES**

Patentes ou pedidos de patente anteriores	Detalhes
US5832130	
US7379496	

GB2274754A1	Samsung Electronics Co. Ltd.
US6529634A1	Thyagarajan
US7676101	
US2006/0204115A1	Berazerovic: utiliza um tipo único de codificação com parâmetros variáveis para blocos codificados

Em geral, muitos codificadores/decodificadores de vídeo não são capazes de codificar eficazmente áreas extensas de imagens com parâmetros substancialmente constantes enquanto sendo simultaneamente capazes de codificar áreas detalhadas altamente espacialmente das imagens. É prática comum utilizar compensação de movimento em uma forma de previsão e métodos de codificação de erro de previsão no uso de transformações, por exemplo, transformação de coseno discreta (DCT) e transformações de ondas pequenas. Estas transformações utilizam um processo em que porções de uma determinada imagem, por exemplo, uma imagem de fotografia ou uma imagem formando uma parte de uma sequência de vídeo, são divididas em blocos que são então submetidos a processos de codificação. Os blocos são, por exemplo, elementos de imagem 8x8, elementos de imagem 4x4 ou similar. Tais blocos relativamente pequenos são empregados porque os tamanhos grandes dos blocos resultam em processos de codificação ineficazes, embora blocos de elementos de imagem 16x16 sejam algumas vezes empregados. De acordo com abordagens contemporâneas conhecidas para codificação de imagem, quando múltiplos tamanhos de blocos diferentes são usados para codificação, é prática comum utilizar uma pequena variação nos tamanhos dos blocos; além do mais, os tamanhos dos blocos são selecionados baseados em quanto bom pode ser o movimento compensado em uma área de bloco associada ou baseados em um parâmetro de qualidade de codificação, por exemplo, um parâmetro de qualidade alvo. Em geral, a qualidade de imagem codificada mais alta requer blocos pequenos que resultem em menos compressão de dados. Certos tipos de codificação contemporânea podem mesmo resultar em um aumento no tamanho dos dados, quando características de correção de erro tais como códigos de paridade e códigos

de correção de erro são incluídos.

A partir do descrito acima, será apreciado que prover compressão de dados de imagens e de vídeos enquanto preservando a qualidade da imagem é um problema contemporâneo que não é endereçado por codificadores e decodificadores conhecidos, a despeito de uma grande variedade de codificadores/decodificadores que foram desenvolvidos durante as décadas recentes.

SUMÁRIO

A presente invenção procura um codificador para codificar dados de entrada representativos de pelo menos um item do conteúdo de dados e gerar dados de saída codificados correspondentes representativos de pelo menos um item do conteúdo de dados, em que os dados de saída codificados são comprimidos com relação aos dados de entrada sem qualquer perda substancial de qualidade que ocorre durante a decodificação; os dados são opcionalmente qualquer tipo de dados, por exemplo, pelo menos um de: dados de imagem, dados de vídeo, dados econômicos, dados de máscara, dados sismográficos, dados convertidos de análogo para digital (ADC), dados de sinais biomédicos, dados texturais, dados de calendário, dados matemáticos, dados binários, mas não limitados aos mesmos.

A presente invenção também procura proporcionar um método para codificar os dados de imagem representativos de pelo menos um item do conteúdo de dados e gerar dados de saída codificados correspondentes representativos do pelo menos um item do conteúdo de dados, em que os dados de saída codificados são comprimidos com relação aos dados de entrada sem qualquer perda substancial de qualidade que ocorre durante a decodificação.

De acordo com um primeiro aspecto, é proporcionado um método de codificar dados de entrada para gerar dados de saída codificados correspondentes como definido na reivindicação 1 anexa: é proporcionado um método de codificar dados de entrada para gerar dados de saída codificados correspondentes, caracterizado em que o método inclui as etapas de:

(a) subdividir os dados de entrada em uma pluralidade de blocos

ou pacotes, os blocos ou pacotes tendo um tamanho dependendo de uma natureza de seu conteúdo, e os blocos ou pacotes sendo de um ou mais tamanhos;

5 (b) aplicar uma pluralidade de transformações ao conteúdo dos blocos ou pacotes para gerar dados transformados correspondentes;

10 (c) verificar uma qualidade de representação dos dados transformados dos blocos ou pacotes comparados ao conteúdo dos blocos ou pacotes antes da aplicação das transformações para determinar se ou não a qualidade de representação dos dados transformados satisfaz um ou mais critérios de qualidade;

(d) em um evento em que a qualidade de representação dos dados transformados do um ou mais blocos ou pacotes não satisfaz o um ou mais critérios de qualidade, subdividir e/ou combinar o um ou mais blocos ou pacotes e repetir a etapa (b); e

15 (e) em um evento em que a qualidade de representação dos dados transformados do um ou mais blocos ou pacotes satisfaz o um ou mais critérios de qualidade, produzindo os dados transformados para prover os dados de saída codificados representativos dos dados de entrada a serem codificados.

20 A invenção é vantajosa em que o método é capaz de proporcionar codificação eficaz dos dados de entrada para prover os dados de saída correspondentes enquanto provendo potencialmente a compressão de dados substancialmente sem perda dos dados de saída codificados relativos aos dados de entrada a serem codificados.

25 Opcionalmente, o método inclui usar as transformações para comprimir o conteúdo associado com os blocos ou pacotes, de modo que os dados de saída codificados são similares em tamanho do que os dados de entrada a serem codificados.

30 Opcionalmente, o método inclui a seleção entre tipos diferentes de transformações para assegurar que um ou mais critérios de qualidade é satisfeito.

Opcionalmente, o método inclui dispor para um ou mais critérios

de qualidade para serem dinamicamente variáveis durante a codificação dos blocos ou pacotes dependendo de uma natureza do conteúdo incluído dentro dos blocos ou pacotes.

5 Opcionalmente, o método é implementado de modo que os blocos ou pacotes são subdivididos e/ou combinados de modo que pelo menos um de seus parâmetros representativos descrevendo seu conteúdo é liso dentro de seus blocos ou pacotes subdivididos e/ou combinados. O pelo menos um parâmetro representativo é, por exemplo, cor, amplitude, resistência, número ou código.

10 Opcionalmente, o método é implementado de modo que os blocos ou pacotes correspondam a uma série de dados, em que a subdivisão dos dados de entrada correspondentes para uma determinada informação formar uma pluralidade de blocos correspondentes é feita dependendo do conteúdo presente em um ou mais dados precedentes à determinada informação dentro da sequência temporal de dados.

15 Opcionalmente, o método inclui adicionar a informação de cabeçalho aos dados transformados na etapa (e) para gerar os dados de saída codificados, em que a informação de cabeçalho inclui informação indicativa das transformações empregadas na etapa (b).

20 Opcionalmente, o método é implementado de modo que a etapa (b) inclui buscar informação suplementar a partir de um arranjo de base de dados para uso quando se executa as transformações, a informação suplementar incluindo pelo menos um dos algoritmos, regras, um ou mais parâmetros de transformação. Mais opcionalmente, o método inclui ainda adicionar a informação de cabeçalho aos dados de saída codificados indicativos do arranjo de base de dados para possibilitar a subsequente decodificação dos dados de saída codificados para acessar a informação suplementar quando se decodifica os dados de saída codificados.

25 Opcionalmente, o método inclui utilizar para as transformações 30 uma ou mais das referências de base de dados, valor DC, cursor, escala, linha, multinível, interpolação, extrapolação, DCT, modulação de código de pulso (PCM), DPCM, RLE, SRLE, EM, LZO, VLC, codificação de Huffman,

codificação aritmética, codificação de faixa, codificação de transformação, codificação delta, codificação ODelta, RLE específico de bzip-2. Outros tipos de transformações também são possíveis de utilizar para o método.

5 Opcionalmente, o método inclui codificar pelo menos um de vídeo, imagem, áudio, gráficos, texto, ECG, sísmico, ASCII, Unicode, e informação binária presentes nos dados de entrada.

10 Opcionalmente, o método inclui codificar múltiplos canais e/ou camadas nos dados de saída codificados para prover pelo menos um de: vídeo interativo, anúncios comerciais, uma pluralidade de pontos de vista durante reportagem de eventos esportivos, marcas d'água interativas, reconhecimento de padrão interativo, botões de interface de usuário 2D/3D animados.

15 Opcionalmente, o método inclui transmitir os dados codificados a pelo menos uma das seguintes destinações: um dispositivo de memória de dados, uma rede de comunicação, um cartão de memória, discos de memória de dados, redes de comunicação de área local (LANs), diretamente a um decodificador.

20 Opcionalmente, a etapa (a) do método inclui dividir inicialmente os dados de entrada (20) em um ou mais blocos baseado em pelo menos um de:

- (a) resoluções de imagem;
- (b) uma quantidade de dados; e
- (c) tamanho de bloco ou pacote máximo.

25 De acordo com um segundo aspecto, é provido um produto de software gravado em meios de armazenamento legíveis por máquina não transitórios, em que o produto de software é executável em hardware de computação para executar um método de acordo com o primeiro aspecto da invenção.

30 De acordo com um terceiro aspecto, é provido uma aplicação de software para um dispositivo de comunicação sem fio móvel, em que a aplicação de software inclui um produto de software de acordo com o segundo aspecto da invenção.

De acordo com um quarto aspecto, é provido um codificador operável para codificar dados de entrada para gerar dados de saída codificados correspondentes, caracterizado em o codificador inclui hardware de processamento de dados que é operável:

5 (a) subdividir os dados de entrada em uma pluralidade de blocos ou pacotes, os blocos ou pacotes tendo um tamanho dependendo de uma natureza de seu conteúdo, e os blocos ou pacotes sendo de um ou mais tamanhos;

10 (b) aplicar uma pluralidade de transformações ao conteúdo dos blocos ou pacotes para gerar dados transformados correspondentes;

(c) verificar uma qualidade de representação dos dados transformados dos blocos ou pacotes comparados ao conteúdo dos blocos ou pacotes antes da aplicação das transformações para determinar se ou não a qualidade de representação dos dados transformados satisfaz um ou mais
15 critérios de qualidade;

(d) em um evento em que a qualidade de representação dos dados transformados do um ou mais blocos ou pacotes não satisfaz o um ou mais critérios de qualidade, para subdividir e/ou combinar, se ela melhora a eficácia de compressão e não deteriora a reconstrução significativamente, o
20 um ou mais blocos ainda e repetir a etapa (b); e

(e) em um evento em que a qualidade de representação dos dados transformados do um ou mais blocos ou pacotes satisfaz o um ou mais critérios de qualidade, para produzir os dados transformados para prover os dados de saída codificados representativos dos dados de entrada a serem
25 codificados.

Opcionalmente, o codificador é implementado de modo que o hardware de processamento de dados é implementado usando hardware de computação operável para executar um produto de software. Mais opcionalmente, o hardware de processamento de dados inclui uma pluralidade de
30 processadores que são operáveis para processar dados em paralelo, e o codificador é operável para dirigir simultaneamente dados de bloco e/ou pacote à pluralidade de processadores para codificar os dados a serem incluí-

dos nos dados de saída codificados.

Opcionalmente, o codificador é operável para selecionar entre tipos diferentes de transformações para assegurar que o um ou mais critérios de qualidade são satisfeitos.

5 Opcionalmente, o codificador é operável para comprimir o conteúdo associado com os blocos ou pacotes, de modo que os dados de saída codificados são menores em tamanho do que os dados de entrada a serem codificados.

10 Opcionalmente, o codificador é operável para dispor o um ou mais critérios de qualidade para serem dinamicamente variáveis durante a codificação dos blocos ou pacotes dependendo de uma natureza do conteúdo incluído dentro dos blocos ou pacotes.

15 Opcionalmente, o codificador é operável para subdividir os blocos ou pacotes de modo que pelo menos um de seus parâmetros representativos descrevendo seu conteúdo é liso dentro de seus blocos ou pacotes subdivididos. Mais opcionalmente, o codificador é implementado de modo que o pelo menos um parâmetro corresponde a uma cor, amplitude, resistência, número ou código dos blocos subdivididos.

20 Opcionalmente, o codificador é implementado de modo que os blocos ou pacotes correspondam a uma série de dados, em que a subdivisão dos dados de entrada correspondentes para uma determinada informação para formar a pluralidade de blocos correspondentes é feita dependendo do conteúdo presente em um ou mais dados precedendo a determinada informação dentro da sequência temporal de dados.

25 Opcionalmente, o codificador é operável para adicionar informação de cabeçalho aos dados transformados para gerar os dados de saída codificados, em que a informação de cabeçalho inclui a informação indicativa das transformações empregadas pelo codificador.

30 Opcionalmente, o codificador é operável para buscar informação suplementar a partir de um arranjo de base de dados para uso quando se executa as transformações, a informação suplementar incluindo pelo menos um dos algoritmos, regras, um ou mais parâmetros de transformação. Mais

opcionalmente, o codificador é operável para adicionar a informação de cabeçalho aos dados de saída codificados de um modo indicativo do arranjo de base de dados para possibilitar a decodificação subsequente dos dados de saída codificados para acessar a informação suplementar quando se decodifica os dados de saída codificados.

Opcionalmente, o codificador é operável para utilizar as transformações de um ou mais de: referência da base de dados, valor DC, cursor, escala, linha, multinível, inalterado, interpolação, extrapolação, DCT, modulação de código de pulso (PCM), DPCM, RLE, SRLE, EM, LZO, VLC, codificação de Huffman, codificação aritmética, codificação de faixa, codificação de transformação, codificação ODelta. Outros tipos de transformação são opcionalmente utilizadas pelo codificador.

Opcionalmente, o codificador é operável para codificar pelo menos um de vídeo, imagem, áudio, gráficos, texto, ECG, sísmico, ASCII, Unicode, e informação binária presentes nos dados de entrada a serem codificados.

Opcionalmente, o codificador é implementado para transmitir os dados codificados a pelo menos uma das seguintes destinações: um dispositivo de memória de dados, uma rede de comunicação, um cartão de memória, discos de memória de dados, redes de comunicação de área local (LANs), diretamente a um decodificador.

De acordo com um quinto aspecto, é provido um produto de consumo eletrônico operável para transmitir e/ou armazenar dados de entrada, caracterizado em que o produto de consumo eletrônico inclui um codificador de acordo com o quarto aspecto para codificar os dados de entrada.

Opcionalmente, o produto de consumo eletrônico é pelo menos um de um telefone móvel, um telefone celular, um computador tablet, um computador tablet, um computador pessoal, uma televisão, um dispositivo de tocadores de mídia portáteis, câmeras, editores, transcodificador, escaners, um fax, uma máquina de copiar, microfones, cartões de áudio, tocadores de gravação, tocadores de DVD, etc.

Será apreciado que as características da invenção são suscetíveis

veis de serem combinadas em várias combinações sem sair do escopo da invenção como definido pelas reivindicações anexas.

DESCRIÇÃO DOS DIAGRAMAS

As modalidades da presente divulgação serão agora descritas, a título de exemplo somente, com referência aos seguintes diagramas, em que:

a FIG. 1 é uma ilustração esquemática de uma modalidade de um codificador;

a FIG. 2 é um fluxograma de etapas de um método de codificar dados de entrada representativos de pelo menos um item do conteúdo de dados para gerar dados de saída codificados correspondentes, em que os dados de saída codificados são comprimidos com relação aos dados de entrada sem perda substancial de qualidade que ocorre durante a codificação; o item do conteúdo de dados inclui benéficamente pelo menos um de: dados de imagem, dados de vídeo, dados de áudio, dados econômicos, dados de máscara, dados sismográficos, dados convertidos de análogo para digital (ADC), dados de sinais biomédicos, dados texturais, dados de calendário, dados matemáticos, dados binários, mas não limitados a esses;

a FIG. 3 é uma partição exemplar de uma imagem em áreas correspondendo a blocos para codificação usando um método cujas etapas são ilustradas na FIG. 2;

a FIG. 4 é uma imagem exemplar e divisão de imagem inicial exemplar para uso de acordo com as modalidades;

a FIG. 5 é um bloco exemplar da imagem exemplar da FIG. 4; e

a FIG. 6 ilustra blocos de dispositivo portátil no qual a codificação é implementada.

Nos diagramas anexos, um número sublinhado é empregado para representar um item sobre o qual o número sublinhado é posicionado ou um item ao qual o número sublinhado é adjacente. Um número não sublinhado refere-se a um item identificado por uma linha ligando o número não sublinhado ao item. Quando um número é não sublinhado e acompanhado por uma seta associada, o número não sublinhado é usado para identificar

um item geral para o qual a seta está apontando.

DESCRIÇÃO DAS MODALIDADES

Em resumo, a presente invenção diz respeito a codificadores e métodos associados de codificação de dados de entrada para gerar dados de saída codificados correspondentes. Os métodos dizem respeito ao recebimento de dados de entrada representativos de uma ou mais imagens e/ou um ou mais sinais de áudio, e então ao processamento dos dados de entrada de um modo correspondente a pelo menos um de:

10 (a) dividir cada imagem em blocos que podem ser de tamanho variável;

(b) dividir o um ou mais sinais de áudio em pacotes que podem ser de tamanho variável;

(c) combinar blocos de imagem para formar blocos grandes que podem ser de tamanho variável que são mais favoráveis para codificar; e

15 (d) combinar pacotes de um ou mais sinais de áudio para formar pacotes combinados que são mais favoráveis para codificar.

Na visão geral, o método inclui o processamento dos blocos e/ou pacotes para gerar dados de saída codificados comprimidos representativos de uma ou mais imagens e/ou de um ou mais sinais de áudio.

20 O método utiliza benéficamente múltiplos métodos de modificação de entropia e de codificação, a saber, transformações, quando codifica e comprime os dados descrevendo blocos de dados. Por exemplo, transformações diferentes são usadas para blocos diferentes de uma determinada imagem, quando desempenho de codificação e compressão favorável é deste modo obtido, dependendo do conteúdo da informação dos blocos diferentes. Beneficamente, a otimização de RD é utilizada como um critério de qualidade quando se processa os blocos de dados. A informação descrevendo a pluralidade de transformações empregadas quando se codifica os blocos é comunicada nos dados de saída codificados; esta informação é tanto intrinsecamente incluída nos dados de saída codificados, ou os dados de saída incluem uma referência a uma ou mais bases de dados para onde a informação descrevendo as transformações utilizadas pode ser obtida. A otimi-

25

30

zação de RD será descrita em maior detalhe posteriormente. Os métodos de codificação que são benéficamente empregados incluem um ou mais de: multinível, linha, escala, interpolação, extrapolação, inalterada, estimativa de movimento, SRLE, EM, Odelta e codificação de faixa, como será elucidado em maior detalhe posteriormente. A divisão de blocos Init (inicialização) também é opcionalmente benéficamente utilizada.

A seleção dos blocos, e/ou combinação dos blocos, é determinada pela facilidade com que as áreas correspondendo aos blocos podem ser codificadas; por exemplo, os blocos maiores são empregadas para áreas de uma ou mais imagens que têm um valor paramétrico associado substancialmente constante, e blocos menores que são empregados para áreas de uma ou mais imagens que são difíceis de codificar tendo por conta as mudanças espaciais relativamente abruptas nos valores paramétricos associados para as áreas. Os parâmetros pertencem, opcionalmente, a uma ou mais de: cor, iluminação, valor de parâmetro de deslizamento, padrão repetitivo, mas não limitados a esses. A codificação fácil corresponde, por exemplo, a pelo menos um parâmetro associado com uma determinada área sendo substancialmente constante dentro de determinada área. Além do mais, o método também utiliza blocos maiores para áreas estacionárias nas sequências de vídeo de imagens, ou para grupos de áreas nas sequências de vídeo de imagens que estão se movendo similarmente, a saber, blocos que correspondem a objetos fixos. Os blocos são opcionalmente retilíneos em relação às áreas da uma ou mais imagens que elas representam, por exemplo, elementos de 64x64, elementos de 32x16, elementos de 4x20, elementos de 10x4, elementos de 1x4, elementos de 3x1, elementos de 8x8, elemento 1x1 e assim por diante; opcionalmente, os elementos correspondem a pixels presentes na uma ou mais imagens, mas podem ser submetidos a operações de escalonamento durante a codificação, a saber, cada elemento correspondendo a uma pluralidade de pixels correspondentes. No entanto, outras formas de blocos são opcionalmente empregadas, por exemplo, blocos elípticos, blocos circulares e assim por diante. Além do mais, por analogia, o método também pode ser aplicado para codificar um ou mais sinais de áudio.

dio, em que o um ou mais sinais de áudio são subdivididos em pacotes, e/ou combinados em pacotes, de comprimento temporal variável, dependendo da natureza dos sinais de áudio correspondendo a esses, e os pacotes são então codificados para gerar os dados de entrada comprimidos codificados; os pacotes são parecidos com os blocos mencionados acima, mas pertencem à informação de áudio em vez de informação de imagens. Os codificadores de acordo com a presente invenção são capazes de codificar simultaneamente tanto informação de áudio como informação de imagem.

Durante o processamento das áreas de uma ou mais imagens em blocos correspondentes, o método inclui verificar uma qualidade de representação de informação provida pelos blocos com relação ao detalhe correspondente na uma ou mais imagens para computar um índice de qualidade correspondente; em um evento em que o índice de qualidade computado indica, quando comparado contra um limiar de qualidade de referência, que uma seleção de tamanhos de bloco foi utilizada de modo que a qualidade de representação de dados providos pelos blocos é insuficiente, o método itera de volta e usa blocos progressivamente menores, e/ou combina vários blocos juntos, se ele promove a eficácia de compressão e não deteriora a reconstrução significativamente, até que o índice de qualidade indique que a qualidade de representação foi encontrada como definido pelo limiar de qualidade de referência. Por tal abordagem é possível, de acordo com a presente invenção, obter a compressão de dados durante a codificação que é substancialmente sem perda, dependendo da escolha do valor limiar para a qualidade de representação de informação. Opcionalmente, o limiar de qualidade de referência torna-se dinamicamente variável, dependendo do conteúdo presente na uma ou mais imagens, por exemplo, quando a uma ou mais imagens são uma parte da sequência de vídeo onde há atividade desordenada rápida, o limiar de qualidade de referência pode ser relaxado durante a atividade desordenada para possibilitar que um grau aprimorado de compressão de dados seja obtido. A atividade desordenada pode ser, por exemplo, aspectos aleatórios tais como fluxo de água turbulenta, chamas, queda de neve, ondas de fumaça, ondas do oceano e assim por diante, em que a

perda de informação não é prontamente discernível quando os dados codificados são substancialmente decodificados em um decodificador.

5 A determinação dos blocos no codificador mencionado acima pode ser opcionalmente baseada em um ou mais critérios como listado na tabela 3.

TABELA 3: SELEÇÃO DE DIVISÃO E/OU SELEÇÃO DE COMBINAÇÃO DE BLOCOS DURANTE A CODIFICAÇÃO DE IMAGEM

Número do critério	Critério
1	Variação ou desvio padrão de dados de bloco como derivado de uma área correspondente de uma imagem de entrada
2	Média ou soma de uma diferença absoluta entre os dados representados por um determinado bloco e uma previsão de seu valor
3	Variação ou desvio padrão de uma diferença absoluta entre os dados representados por um determinado bloco e uma previsão de seu valor

Opcionalmente, as previsões na tabela 3 são baseadas em regras conhecidas empregadas quando se codifica uma ou mais imagens. Alternativamente, as previsões na tabela 3 são baseadas em informação de configuração provida, por exemplo, como provida a partir de referências de base de dados selecionadas, de direção de previsão, de movimentos do bloco coordenado dentro de uma ou mais imagens, e assim por diante. Um uso de uma variação ou um desvio padrão é uma abordagem utilizada de acordo com a presente invenção para prover a compressão de informação descrevendo uma relação mútua de elementos incluídos dentro de um determinado bloco correspondente. Em muitas situações, as previsões de dados de bloco com a codificação associada são por si mesmas suficientes quando realizando a codificação de acordo com a presente invenção, mas é opcionalmente desejável incluir dados de erro de previsão de código dentro da previ-

são para melhorar uma precisão da previsão. Em uma modalidade simples da presente invenção, um método de previsão de dados simples é empregado, por exemplo, um valor médio, a saber, valor "DC", de pixels ou elementos dentro de um dado bloco a ser liberado nos dados de saída codificados.

5 A divisão de áreas, alternativamente a combinação de áreas, de uma ou mais imagens providas como dados de entrada a um codificador implementando o método mencionado acima é opcionalmente implementada de acordo com qualquer modo que proporcione tanto compressão como também mantém substancialmente a qualidade da imagem, a saber, é substancialmente sem perda durante a codificação. O método aplica várias estratégias para tal divisão e/ou combinação de áreas. Por exemplo, se um determinado bloco inclui informação considerável, se ele é opcionalmente dividido em uma pluralidade de blocos pequenos correspondentes que são relativamente "lisos", alternativamente, opcionalmente combinados em blocos maiores que são relativamente "lisos", a saber, substancialmente constantes, em relação a seu conteúdo de modo que eles individualmente incluem informação relativamente pequena. Quando o método de codificação de acordo com a presente invenção é aplicado a pelo menos uma ou mais imagens e/ou um ou mais sinais de áudio, a qualidade da codificação e o ruído da codificação nos dados de saída codificados são opcionalmente empregados para controlar um modo no qual a divisão das imagens de entrada e sinais de entrada de áudio nos blocos e pacotes, respectivamente, ocorre. No entanto, será apreciado que outros tipos de itens de conteúdo de dados podem ser processados de um modo similar, por exemplo, pelo menos um de:

10 dados de imagem, dados de vídeo, dados de áudio, dados econômicos, dados de máscara, dados sismográficos, dados convertidos de análogos para digitais (ADC), dados de sinais biomédicos, dados texturais, dados de calendário, dados matemáticos, dados binários, mas não limitados a esses.

 Opcionalmente, o ruído no dados codificados de saída codificados é baseado em pelo menos um de:

30

- (i) ruído presente em um bloco ou pacote presente;
- (ii) ruído presente em um ou mais blocos ou pacotes anteriores

pelo método; e

(iii) imagens anteriores.

Opcionalmente, quando uma determinada imagem de entrada é dividida em áreas e blocos correspondentes, o método analisa os blocos gerados deste modo para determinar se ou não qualquer um dos blocos pode ser combinado junto, como mencionado acima, submetido aos critérios de qualidade mencionados acima, a fim de obter um grau maior de compressão de dados nos dados de saída codificados. No mencionado acima, os dados de saída codificados incluem informação associada com os blocos que definem os locais de suas áreas correspondentes em suas imagens originais nos dados de entrada a serem codificados.

Quando se codifica a uma ou mais imagens presentes nos dados de entrada para serem codificados usando o método, os dados associados com as imagens de entrada são benéficamente reduzidos, por exemplo, reduzidos nas relações de 2x1: 1,2x2: 1,1x2: 1,4x1:1, ou similarmente quantificados antes de serem submetidos aos métodos de codificação mencionados acima. Opcionalmente, tal redução é realizada em resposta a uma qualidade desejada de codificação desejada nos dados de saída codificados comprimidos gerados a partir de métodos de aplicação de acordo com a presente invenção. Opcionalmente, os blocos maiores processados pelos métodos são menos quantificados do que os blocos menores; em outras palavras, um grau de quantificação empregado é opcionalmente diminuído na medida em que os tamanhos dos blocos são aumentados. Opcionalmente, durante a codificação, o fator de escalonamento para a redução utilizada, torna-se dinamicamente variável, por exemplo, em resposta a uma natureza do conteúdo em uma sequência de imagens, por exemplo, vídeo, a ser codificada.

Durante a codificação de blocos de acordo com o método, cada bloco tem vários parâmetros que descrevem seus conteúdos. Estes parâmetros são transportados quando se codifica através de vários "canais". Por exemplo, os canais coloridos descrevendo os blocos de uma imagem podem incluir um ou mais de: branco/preto (B/W), Y, U, V, vermelho (R), azul (B), ciano (C), magenta (M), Y e K. Além do mais, as imagens de entrada para

codificação e os blocos podem ser opcionalmente processados quando se executa os métodos usando uma variedade de cores potenciais e formatos de pixels, por exemplo, padrões e formatos contemporâneos Y, YUV420, YUV422, YUV444, RGB444, G e CMYK. Além do mais, o formato é opcionalmente planar, linha intercalada planar e assim por diante. Além do mais, os métodos da invenção são benéficamente operados para trocar o formato das imagens e/ou dos blocos quando realizando atividades de codificação; por exemplo, uma imagem original está em um formato RGB intercalado e é codificada usando os métodos de acordo com a invenção para gerar dados de saída codificados no formato YUV420 ou vice versa.

A profundidade de bit, a saber, a faixa dinâmica de um pixel quando implementando o método de codificação mencionado acima, está benéficamente em uma faixa de resolução de 1-bit a 64-bit. Opcionalmente, cores de pixels diferentes ou canais de áudio podem estar codificando com resoluções mutuamente diferentes, com a condição de que os critérios de qualidade de codificação e o desempenho de compressão dos métodos de codificação sejam satisfeitos.

Os métodos de codificação de acordo com a presente invenção são opcionalmente implementados usando parâmetros de codificação e regras de codificação e/ou tabelas que são armazenados em uma base de dados e que são avaliados quando realizando as atividades de codificação. Opcionalmente, a base de dados está aumentando durante o processo de codificação e liberada para uso quando implementando o método através de um codificador. Por exemplo, a compensação de movimento durante a codificando é benéficamente implementada usando bases de dados liberadas de informação para o codificador. Benéficamente, o codificador é operável para codificar a informação de pixel original presente nos dados de entrada e/ou codificar a informação de erro de previsão. O uso de informação da base de dados quando codificando os dados de entrada para gerar dados de saída codificados correspondentes possibilita que o codificador adapte-se às revisões nos padrões de codificação de parâmetros, tabelas e similares utilizados para codificação. As abordagens de codificação que podem ser adota-

das quando se implementa os métodos de codificação de acordo com a presente invenção opcionalmente incluem um ou mais de: referência de base de dados, valor DC, cursor, escala, linha, multinível, inalterado, interpolação, extrapolação, DCT, modulação de código de pulso (PCM), DPCM, RLE, SRLE, EM, LZO, VLC, codificação de Huffman, codificação aritmética, codificação de faixa, codificação de transformação, codificação delta, codificação ODelta, RLE específico de bzip-2. Opcionalmente, as abordagens de codificação que incluem qualquer combinação dos exemplos mencionados acima de codificação, a saber, uma pluralidade de transformações de codificação é

5
10
15
20
25
30

beneficamente utilizada, bem como uma pluralidade de parâmetros, para blocos e/ou pacotes, dependendo do conteúdo da informação dos blocos e/ou pacotes. Quando uma abordagem de codificação tal como codificação de Huffman é utilizada, tal codificação usa beneficamente tabelas fixas de parâmetros de codificação ou tabelas liberadas de parâmetros de codificação. O codificador é beneficamente implementado usando hardware de computação tendo arranjos de armazenamento de dados, em que tabelas otimizadas de parâmetros de codificação podem ser armazenadas nos arranjos de armazenamento de dados para uso futuro quando realizando operações de codificação. Beneficamente, os endereços de referência para possibilitar que um decodificador acesse as bases de dados para obter os parâmetros apropriados para decodificar os dados de saída codificados a partir do codificador são incluídos nos dados de saída codificados. Opcionalmente, as bases de dados são acessíveis através de uma rede de comunicação, por exemplo, via Internet. Opcionalmente, as bases de dados são suportados através de arranjos de computação em nuvem. Quando o método implementado no codificador utiliza bases de dados geradas matematicamente, as bases de dados podem, opcionalmente, ser o valor DC, transição linear 1D/2D, transição de curva 1D/2D, uma função de transformação 1D/2D ou alguma estrutura de pacote de áudio ou bloco de imagem conhecida.

O método de codificação de acordo com a presente invenção, quando executado em um codificador, é operável para codificar os dados de entrada para gerar dados de entrada codificados, em que os dados de saída

codificados podem ser produzidos como um fluxo de bits, alternativamente, armazenados em meios de armazenamento de dados, por exemplo, como um arquivo de dados. Além do mais, o método da invenção é capaz de ser utilizado em uma faixa de aplicações possíveis; benéficamente, um cabeçalho para vídeo, imagem, bloco de imagem, áudio ou pacotes de áudio inclui informação suplementar, tais como número da versão, tamanho de dados para o vídeo, a imagem ou o pacote, limiar de fator de qualidade empregado quando codificando, tamanho de bloco ou pacote máximo, abordagens de codificação aplicadas, a saber, tipos de transformações empregadas, tabelas de parâmetros de codificação, e qualquer outra informação para auxiliar os processos de decodificação subsequentes. Opcionalmente, a informação que não varia entre os blocos não está incluída para obter um grau aprimorado de compressão de dados nos dados de saída de codificação, ou está incluída em um nível mais alto nos dados de saída codificados, por exemplo, em nível de cabeçalho ou subcabeçalho. A tabela 4 provê uma ordem hierárquica de níveis que são benéficamente empregados nos dados de saída codificados gerados pelo codificador.

TABELA 4: ORDEM DE NÍVEIS NOS DADOS DE SAÍDA CODIFICADOS, DE ALTO PARA BAIXO

Ordem de nível	Informação associada com nível
Alto	Vídeo
	Grupos de imagem
	Imagem
	Grupos de macroblocos
Médio	Macro blocos
	Grupos de blocos
	Bloco
	Grupos de microblocos
Baixo	Microblocos

sente invenção é operável quando executado para selecionar e liberar informação pertencendo a um ou mais níveis nos dados de saída codificados, por exemplo, dependendo do campo de aplicação do método, por exemplo, produtos de vídeo de consumo, aparelho de compressão de imagem profissional para uso em pesquisa, raios X, aparelho de formação de imagem, 5 aparelho de formação de imagem magnética (MRA). Considerações similares pertencem a ordens de níveis nos dados de entrada codificados quando o método de acordo com a presente invenção é empregado para codificar dados de áudio; podem haver cabeçalhos empregados para áudio, grupo de 10 pacotes, pacotes, subpacotes, grupos de segmentos de formas de onda, e segmento de formas de onda.

Com referência à FIG. 1, é mostrada uma ilustração de um codificador de acordo com a presente invenção. O codificador é indicado por 10 e é operável para receber dados de entrada 20 e para codificar os dados de 15 entrada 20 empregando um método de codificação de acordo com a presente invenção para gerar dados de saída 30 comprimidos codificados correspondentes. Os dados de saída 30 são benéficamente codificados em um modo substancialmente sem perda como mencionado acima. Opcionalmente, o codificador 10 é acoplado através de uma rede de comunicação 40 a 20 um arranjo de base de dados 50 na qual um ou mais parâmetros, tabelas e/ou regras para codificar os dados de entrada 20 são armazenados.

Em operação, o codificador 10 recebe os dados de entrada 20, opcionalmente deriva a informação de codificação a partir do arranjo de base de dados 50 através de uma rede de comunicação 40, e então prossegue 25 para codificar os dados de entrada 20 para gerar os dados de saída 30 comprimidos codificados. Opcionalmente, os dados de entrada 20 incluem pelo menos um de: áudio, uma ou mais imagens, vídeo, gráficos, texto, ECG, sísmicos, ASCII, Unicode, e dados binários, mas não estão limitados a esses. Opcionalmente, os dados de saída codificados 30 incluem cabeçalhos, 30 informação de codificação bem como dados codificados. Os dados de saída 30 podem ser transmitidos a partir do codificador 10, por exemplo, para comunicação através de um arranjo de rede de comunicação a um ou mais

dispositivos de armazenamento de dados ou decodificadores, ou armazenados diretamente em meio de armazenamento de dados legíveis por máquina, por exemplo, de armazenamento de unidade de disco rígido do servidor, dispositivos de memória em estado sólido portáteis e assim por diante.

5 O codificador 10 é benéficamente implementado como hardware, por exemplo, através de um ou mais PGLA (Matriz Lógica de Portal Programável), através de uma ou mais aplicação de software de hardware de computação, ou qualquer mistura de hardware e software, por exemplo, uma máquina de processamento paralelo. O codificador 10 pode ser empregado
10 em produtos multimídia, computadores, telefones móveis ("telefones celulares"), serviços de Internet, gravadores de vídeo, tocadores de vídeo, aparelhos de comunicação e similares. O codificador 10 é opcionalmente empregado em conjunto com sistemas de captura de imagem, por exemplo, câmaras de vigilância, sistemas de raios X em hospital, escaners MRI em hospital,
15 escaners de ultrassom em hospital, sistemas de vigilância aérea e aparelhos similares que geram grandes quantidades de dados de imagem em que compressão sem perda é desejado de modo a preservar a informação fina das imagens enquanto tornando as quantidades dos dados de imagem controláveis para fins de armazenamento de dados.

20 O codificador 10 é utilizável benéficamente com aparelho conhecido para processamento de imagem, por exemplo, em conjunto com um processador de multi/vídeo como descrito no pedido de patente US publicado nº US2007/280355 incorporado ao presente documento por referência, por exemplo, em conjunto com um gerador de imagem como descrito no
25 pedido de patente US publicado nº US2010/0322301 incorporado ao presente documento por referência, e, por exemplo, com um reconhecedor de padrão como descrito em um pedido de patente US publicado nº US2011/007971 incorporado ao presente documento por referência.

Um método de codificar dados de entrada usando o codificador
30 10 da FIG. 1 será agora descrito com referência à FIG. 2. Na FIG. 2, as etapas de um método de codificação de dados de entrada 20 são indicadas por 100 a 200. Em uma primeira etapa 100, o método inclui o recebimento dos

dados de entrada para os blocos, imagem, vídeo e/ou áudio, mas não limitados a esses, por exemplo, para receber os dados de entrada 20 mencionados acima; a informação de tipo de imagem/bloco, de formato de imagem/bloco é derivada na primeira etapa para inclusão nos dados de saída codificados 30. A primeira etapa 100 é capaz de prover informação de cabeçalho escrita, por exemplo, informação de tamanho de imagem, nos dados comprimidos codificados.

Em uma segunda etapa 110, executada depois da primeira etapa 100, o método inclui criar, se necessário, blocos ou pacotes iniciais. Em uma terceira etapa 120, executada depois da segunda etapa 110, o método inclui a realização de uma análise da informação de blocos e/ou pacotes gerada pela segunda etapa 110, Em uma quarta etapa 130, executada depois da terceira etapa 120, o método inclui determinar se ou não um ou mais blocos e/ou pacotes identificados na terceira etapa 120 precisa ser dividido e/ou combinado, por exemplo, para obter uma qualidade definida da relação de codificação e/ou compressão definida, a saber, codificação substancialmente sem perda; em um evento em que um ou mais blocos ou pacotes é encontrado para divisão e/ou combinação necessária na quarta etapa 130, o método prossegue para a quinta etapa 140 em que o um ou mais blocos ou pacotes identificados são divididos em blocos ou pacotes menores, e/ou são combinados em blocos ou pacotes maiores, deste modo criando mais blocos novos ou pacotes novos; em um evento em que um ou mais blocos ou pacotes não são requeridos para serem divididos e/ou combinados na quarta etapa 130, o método prossegue para a sexta etapa 150. Ao término da quinta etapa 140, quando apropriado, o método prossegue para a sexta etapa 150.

Na sexta etapa 150, os dados dos blocos reconstruídos pelas transformações com base na análise executada na etapa 120 são emitidos para as correntes de dados, após o que o método prossegue ainda para uma sétima etapa 160.

Na sétima etapa 160, o método inclui determinar se ou não um último bloco ou pacote foi alcançado, que deve ser codificado; em um evento

em que existem blocos ou pacotes restantes para serem codificados, o método inclui retornar à terceira etapa 120 de realizar a análise da informação de bloco ou pacote; em um evento em que todos os blocos ou pacotes foram codificados, o método inclui prosseguir para uma oitava etapa 170. A oitava

5 etapa 170 do método inclui a compressão ou codificação de dados, por exemplo, valores, parâmetros e informação de divisão de pixéis, por exemplo, usando transformações RLE, de Huffman, DCT ou similares, e informação de cabeçalho escrita, por exemplo, se os valores DC dos blocos são enviados, eles podem ser primeiramente opcionalmente codificados por del-

10 ta, e depois os valores codificados por delta como RLE e então estes codificados usando codificação de Huffman. Quando as atividades de compressão e codificação na oitava etapa 170 foram concluídas, o método inclui prosseguir para uma nona etapa 180 que diz respeito à verificação de se ou não o último bloco, pacote ou quadro inicial foi alcançado; em um evento em

15 que o pelo menos um bloco, pacote ou quadro inicial não foi alcançado, o método inclui retornar à terceira etapa 120; em um evento em que o último bloco, pacote ou quadro inicial foi alcançado, o método inclui prosseguir para uma décima etapa 190. Na décima etapa 190, o método inclui realizar uma ou mais análises se ou não imagens/blocos são, por exemplo, similares às

20 imagens/blocos anteriores ou, por exemplo, na cor preta. Tal análise opcionalmente também necessita de alguma informação que pode ser obtida a partir de uma ou mais das etapas anteriores. Esta etapa possibilita que as imagens/blocos dos dados de saída codificados sejam mais comprimidas. Análises similares que não precisam de qualquer informação a partir de uma

25 ou mais das etapas anteriores também são executadas já na terceira etapa 120, e então também é possível evitar o processamento desnecessário para ser executado no processador. Depois de concluir a décima etapa 190, o método inclui o prosseguimento para uma décima primeira etapa 200, em que o método inclui aplicar a codificação final do bloco, pacote, imagem ou

30 vídeo para finalizar os dados de saída 30 comprimidos codificados. Opcionalmente, o método inclui prosseguir da sétima etapa 160 diretamente para a décima primeira etapa 200 se aplicável, por exemplo, somente um bloco

na imagem é necessário para ser codificado e ele é, por exemplo, preto ou similar do que o bloco anterior na imagem anterior. Todos os dados codificados que podem ser escritos para um arquivo ou de transmitidos para fora são benéficamente gerados tão antes quanto possível para evitar armazenamento temporário extra quando implementando o método.

Com referência à FIG. 3, a partir do descrito acima, será apreciado que o método de codificação de acordo com a presente invenção utiliza, quando apropriado, tamanho de bloco ou pacote variável para proporcionar uma solução ótima entre a compressão de dados nos dados de saída codificados 30 e compressão substancialmente sem perda, a saber, substancialmente sem perda discernível. Na FIG. 3, bloco de codificação 300 grandes são empregados para um canto do lado direito superior de uma determinada imagem, em que blocos menores 310, 320, 330 são requeridos ao longo do lado direito e áreas da borda inferior da imagem para prover mais precisamente a codificação destas áreas. Nos dados de saída codificados 30, parâmetros descrevendo o conteúdo da imagem dos blocos 300, 310, 320, 330 e a posição dos blocos dentro da imagem são incluídos nos dados de saída codificados 30. Benéficamente, os métodos de codificação empregados para codificar os blocos 300, 320, 320, 330 também são definidos nos dados de saída codificados 30, por exemplo, métodos diferentes para os blocos diferentes 300, 310, 320, 330. Distribuição dos blocos 300, 310, 320, 330 irá variar dependendo da distribuição espacial do conteúdo dentro das imagens a serem codificadas. A FIG. 3 mostra um exemplo da divisão inicial dos blocos na imagem que é gerada na segunda etapa 110. Esta divisão inicial de blocos não requer qualquer informação a ser enviada entre o codificador e o decodificador, porque ela pode ser baseada, por exemplo, em um tamanho da imagem. De acordo com uma modalidade, quando uma divisão real de bloco é executada na quinta etapa 140, então essa informação pode ser liberada a partir do codificador para o decodificador. Um exemplo prático na codificação de uma imagem será descrito depois.

No descrito acima, as seguintes abreviações foram usadas como dada na tabela 5, Estes vários formatos de codificação são todos potencial-

mente relevantes para uso quando se implementa o codificador 10, dependendo do desempenho desejado do codificador 10.

Referindo-se novamente à FIG. 2, será apreciado que o método de codificação opcionalmente completa a codificação de pelo menos uma das imagens, vídeo e áudio, e então depois produz os dados de saída codificados 30. Alternativamente, por meio de uma ligação após a etapa 200 na FIG. 2 de volta à etapa 100 ou 120, possibilita que os dados de saída codificados 30 sejam produzidos, por exemplo, transmitidos, simultaneamente com o método de execução das etapas de codificação em pelo menos uma das imagens, vídeo e áudio. Tal operação é benéfica quando o método da presente invenção é empregado para codificar os dados da fonte em um local no servidor em tempo real para transmitir aos clientes, por exemplo, serviços de multimídia liberados pela Internet. O método é deste modo potencialmente capaz de funcionar com capacidade de armazenamento de dados pequena disponível, e a interrupção do usuário dos dados codificados transmitidos, por exemplo, pelo usuário selecionando uma alternativa, evita que todo um arquivo de vídeo seja codificado quando não requerido pelo usuário, deste modo economizando os recursos de processamento do computador. Tais considerações são muito importantes para os serviços de transmissão de multimídia contemporâneos via Internet, por exemplo.

Quando se envia os dados de saída codificados 30, por exemplo, em uma situação de codificação/transmissão simultânea como mencionado acima, é benéfico, por exemplo, que cabeçalhos de nível mais alto sejam enviados primeiro e depois os dados codificados referentes a um nível mais alto da hierarquia, depois os dados codificados referentes aos níveis mais baixos na hierarquia são enviados. Por exemplo, nos dados de saída codificados 30, os dados de tamanho de imagem e dados referentes a imagem/vídeo são enviados primeiro nos dados de saída codificados 30, e depois a informação referente à divisão dos blocos, e depois algoritmos utilizados na codificação do bloco e os dados de bloco codificados.

O codificador 10 é implementado benéficamente usando hardware de computação incluindo uma arquitetura de processador paralela

que inclui uma pluralidade de processadores organizados para processar dados em paralelo. Por exemplo por conta da divisão de blocos inicial para prover os blocos de divisão 300, 310, 320, 330, os blocos divididos 300, 310, 320, 330 podem ser distribuídos para a pluralidade de processadores, por exemplo, um primeiro processador codificando os blocos 300, 310, 320 e um segundo processador codificando os blocos 330.

Durante a codificação, o número de canais empregados pelo codificador 10 para codificar imagens, vídeo e/ou áudio pode ser dinamicamente variável, para obter um grau maior de compressão de dados nos dados de saída codificados 30. Por exemplo, é conveniente utilizar até quatro canais, embora as imagens possam incluir múltiplas camadas que podem ser apresentadas juntas. As camadas, opcionalmente, pertencem a um ou mais de: subtítulos, logos, imagens de múltiplas vistas, profundidade. Além do mais, as camadas opcionalmente transportam mutuamente volumes diferentes de dados. Todos os dados de camada associados são codificados nos dados de saída codificados 30 como canais separados, e uma descrição associada com respeito a como os canais devem ser combinados e usados é também incluída em um ou mais cabeçalhos incluídos nos dados de saída codificados 30.

TABELA 5: ABREVIACÕES PARA TRANSFORMAÇÕES DE CODIFICAÇÃO UTILIZÁVEIS QUANDO IMPLEMENTANDO AS MODALIDADES DA PRESENTE INVENÇÃO

1D	1-Dimensional (por exemplo, para um pacote de sinais)	MAD	Diferença absoluta média
2D	2-Dimensional (por exemplo, para um bloco, imagem, áudio estéreo ou de múltiplos canais)	MP3	Camada de áudio 3 MPEG-1
3D	Tridimensional (por exemplo, para vídeo, imagem estéreo, imagem de múltiplos canais)	MPEG	Grupo de peritos de fotografia em movimento

AAC	Codificação de áudio avançada		MSD	Diferença de quadrado média
AVC	Codificação de vídeo avançada		MVC	Codificação de vídeo de múltiplas vistas
BMP	Bitmap – formato de arquivo		PCM	Modulação de código de pulso
DC	Corrente direta		PNG	Gráficos de rede portáteis
DCT	Transformação de cosseno discreta		RLE	Codificação de comprimento de execução
DPCM	Modulação de código de pulso diferencial		SAD	Soma das diferenças absolutas
FLAC	Codificador/decodificador de áudio sem perda livre		SSD	Soma das diferenças de quadrado
GIF	Formato de intertroca de gráficos		TIFF	Formato de arquivo de imagem etiquetada
JPEG	Junta de grupo de peritos fotográficos		VLC	Codificação de comprimento variável
JPEG XR	Faixa estendida JPEG		VQ	Quantificação de vetor
LZO	Método de codificação baseado na transformação Lempel-Ziv		EM	Modificador de entropia

5 Será apreciado que a divisão de blocos ilustrada na FIG. 3 é simplesmente um exemplo. Em uma sequência temporal de imagens, por exemplo, uma sequência de vídeo, verifica-se que os objetos representados nas imagens com bordas bem definidas frequentemente se movem em torno para dentro de um determinado campo de visão das imagens, resultando na divisão de blocos após os movimentos dos objetos. Nas imagens fotográficas a divisão de blocos cria tipicamente o resultado que segue as bordas na imagem espacial. Quando a divisão de blocos é ilustrada a partir da imagem fotográfica os objetos podem frequentemente ser detectados a partir da

mesma.

O método de codificação de acordo com a invenção, por exemplo, como representado na FIG. 2, é capaz, através da camada e da codificação de canal executada no codificador 10, capaz de suportar as apresentações de vídeo interativas para prover novos tipos de serviços de liberação de conteúdo, por exemplo, anúncios comerciais interativos, perspectivas de visualização diferentes quando transmitindo atividades esportivas ao vivo tal como a fórmula 1 e assim por diante. Por exemplo, o codificador 10 permite filmes com camadas de subtítulo localizadas, marcas d'água interativas, reconhecimento de padrão interativo, botões de interface de usuário (UI) 2D/3D animada e assim por diante.

Exemplo simplificado de uma modalidade de codificação é mostrado na FIG. 4. Uma imagem 400 a ser codificada tem uma área azul lisa 414, mostrada como branca na FIG. 4 para clareza, uma área verde 412, mostrada pontilhada na FIG. 4 para clareza, e uma área vermelha de deslocamento 410, mostrada como sombreamento pontilhado deslizante na FIG. 4 para clareza. O tamanho da imagem 400 pode ser arbitrário, por exemplo, elementos ou pixéis 12x8, ou elementos ou pixéis 24x16, ou elementos ou pixéis 120x80, e assim por diante. Os tamanhos de bloco iniciais 402 para a imagem 400 são selecionados como elementos ou pixéis 4x4, ou elementos ou pixéis 8x8, ou elementos ou pixéis 40x40, respectivamente, para tamanhos de imagem diferentes. Na modalidade, existem seis blocos de elementos ou pixéis A, B, C, D, e F como ilustrado. De acordo com a modalidade, o primeiro bloco A é analisado. O bloco A é determinado a partir de análise para ser uniforme e consistindo de azul somente, com o valor de cor representado pelo número 10. O bloco A é considerado como uma área. O bloco B é analisado em seguida. O bloco B também consiste de azul somente, portanto, ele é considerado como uma área. O bloco C é analisado em seguida. O bloco C não é uniforme, portanto, é dividido para quatro sub-blocos indicados por C1, C2, C3 e C4. O primeiro sub-bloco C1 é analisado. Ele consiste de uma área vermelha deslizante, com um valor de cor de 15 em seu canto esquerdo inferior, valores de 20 nos cantos esquerdo de topo e direito de

fundo do sub-bloco C1, e um valor de 25 no canto direito de topo do sub-bloco C1. O sub-bloco C1 é considerado como uma área. Os sub-blocos C2, C3 e C4 são analisados, respectivamente. Os sub-blocos C2, C3 e C4 consistem de azul liso e são, cada um, indicados como áreas azuis lisas. O bloco D é analisado em seguida. O bloco D tem somente uma cor, a saber, azul liso, e é considerado como uma área. O bloco E é analisado em seguida. O bloco E não é uniforme e é assim dividido em quadro sub-blocos E1, E2, E3 e E4. Os sub-blocos E1 e E2 são ambos uniformes com uma cor azul liso. O sub-bloco E3 não é uniforme e é dividido ainda em quatro sub-blocos E31, E32, E33 e E34. Cada sub-bloco E31, E32, E33 e E34 é analisado. O sub-bloco E31 é analisado para consistir de cor uniforme de azul e é marcado como uma área azul lisa. Os sub-blocos E32, E33 e E34 são analisados para serem verde, com um valor de cor de 5 como um exemplo, e são projetados para serem verdes. O sub-bloco E4 é analisado e determinado ser uniforme e projetado como a cor verde. O bloco F é analisado e determinado não ser uniforme. O bloco F é dividido em quatro sub-blocos F1, F2, F3 e F4. O sub-bloco F1 é analisado e determinado não ser uniforme. O sub-bloco F1 é dividido ainda nos sub-blocos F11, F12, F13 e F14. Cada sub-bloco F11, F12, F13 e F14 é analisado. O sub-bloco F11 é determinado ser uniforme com uma cor azul. Os sub-blocos F12, F13 e F14 são determinados ser uniforme com uma cor verde. Os valores são associados com blocos e sub-blocos, respectivamente. Os sub-blocos F2, F3 e F4 são ainda analisados e determinados ser uniformes com uma cor verde. Os valores são associados com os blocos e sub-blocos, respectivamente.

Um exemplo de implementação de código para o processo de codificação mencionado acima é dar um valor "0" para um bloco que não é dividido, e um valor "1" para um bloco que precisa ser dividido. Usando essa lógica, o exemplo acima é beneficentemente codificado como os seguintes bits divididos:

0 0 1 0000 0 1 0010 0000 1 1000 0000,

Em que os primeiros dois valores lógicos "0" ilustram que os dois blocos iniciais A, B não são divididos, o terceiro bloco C é dividido e definido

por um valor lógico "1" como mencionado acima, mas os sub-blocos não são divididos, o quarto bloco D não é dividido, o quinto bloco E é dividido, os primeiros dois, os primeiros dois sub-blocos E1, E2 não são divididos, mas os sub-blocos E3 é dividido e assim por diante. A cadeia de bits ainda pode ser, por exemplo, o comprimento codificado de execução.

Outro método de "preenchimento" para cada bloco pode ser codificado, por exemplo, com um valor lógico de "0" para atribuir a cor lisa para o bloco, e um valor lógico de "1" para atribuir cor de deslizamento para o bloco, referido como Bits do Método:

10 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

No código, os primeiros dois blocos A, B são de cor lisa, o bloco seguinte C1 está deslizando e um restante são cores lisas. Os Bits do Método podem ainda ser, por exemplo, comprimento codificado de execução.

Os valores de cor, respectivamente, para os blocos são referidos como Valores, como a seguir:

10 10 20 25 15 20 10 10 10 10 10 10 5 10 5 5 5 5 5 5 10 5 5 5

em que um conjunto "20, 25, 15, 20" descreve os valores de cor de cada canto do bloco C1. A cadeia de Valores pode ser, por exemplo, codificada por Huffman para reduzir o tamanho dos dados. Além do mais, uma vez que os grupos dos sub-blocos E31, E32, E33 e E34 e o grupo de sub-blocos F11, F12, F13 e F14 têm a mesma combinação e ordem dos valores de cor, a saber, (10, 5, 5, 5), eles podem ser considerados como um elemento e projetar seu próprio valor, referido como Valor de Combinação. O Valor de Combinação pode ser armazenado em uma base de dados, por exemplo, do codificador ou um decodificador correspondente, e referido como um número de Identificação de Referência quando necessário. Frequentemente é benéfico usar fluxos separados para valores de métodos diferentes, por exemplo, no exemplo descrito acima em que os valores DC e os valores de cursor podem ser definidos para separar os fluxos. Alguns métodos, opcionalmente, também geram múltiplos fluxos por si mesmos para possibilitar uma entropia menor para cada fluxo que possibilite melhor eficácia de compressão, nos métodos de codificação de entropia, por exemplo, usando um método de

múltiplos níveis em que níveis altos e baixos usados quando implementando o método podem ser definidos para processar fluxos separados, porque frequentemente os valores altos estão mais próximos uns dos outros, similarmente como os valores baixos estão mais próximos uns dos outros, e então

5 a codificação da faixa que usa valores baixos e altos codificados por delta opera eficazmente.

De acordo com as modalidades Bits Divididos, Bits de Método, Valores e Valores de Combinação podem ser armazenados em hardware de codificador, por exemplo, para suprimento subsequente a outro hardware

10 para fins de decodificação.

Em uma modalidade, a codificação pode ser implementada em um dispositivo portátil 600 da FIG. 6, tal como um telefone inteligente, uma câmera digital ou uma câmera de vídeo. O dispositivo portátil 600 inclui opcionalmente uma câmera (KAM) 604 para capturar uma ou mais imagens,

15 um monitor (DISP) para mostrar a uma ou mais imagens, um receptor/transmissor (RX/TX) 606 para possibilitar comunicação usando redes sem fio celulares ou redes de área local, e outra Entrada/Saída (I/O) 612 tal como Barramento em Série Universal (USB) ou Ethernet, uma Unidade de Processamento Central (CPU) 608 para executar os algoritmos e instruções

20 relativos ao codificador, e uma memória (MEM) 610 para armazenar a uma ou mais imagens a partir da câmera 604, software para o codificador e conteúdo de imagem codificado. O dispositivo portátil 600 é benéficamente configurável para armazenar uma ou mais imagens codificadas em sua memória local 610, e/ou pode ser configurado para enviar periodicamente, quando em

25 solicitação, na ação do usuário, em tempo real ou substancialmente em tempo real, imagens codificadas através do receptor/transmissor (RX/T) 606 ou através de Entrada/Saída (I/O) 612 para sistemas externos.

No descrito acima, a otimização de RD é utilizada para minimizar um valor RD que é derivado da equação 1 (Eq. 1) como a seguir:

30
$$RD = (E) + (\lambda \dots (B_c)) \quad \text{Equação 1}$$

em que

$$E = \text{erro}$$

λ = um coeficiente definindo um custo de uma contagem de bit como uma função do erro E;

B_c = contagem de bit.

O erro E pode ser, por exemplo, MAE, MSE, MAX ou alguma
5 combinação ponderada de tais parâmetros, por exemplo, MAE+MAX. O parâmetro λ refere-se ao coeficiente mencionado acima com o qual um custo da contagem de bit BC pode ser feito comparável com respeito ao erro E, o parâmetro λ torna-se maior conforme menos bits são usados para codificação, em que mais erros são permitidos, a saber, em uma situação onde surge
10 uma grande compressão de dados correspondendo a um valor de critério de qualidade menor. De modo correspondente, o parâmetro λ torna-se menor conforme mais bits são usados para codificação onde menos erro é permitido, a saber, em uma situação onde surge uma pequena compressão de dados correspondendo a um valor de critério de qualidade maior.

15 Com respeito ao critério de qualidade de codificação, ele pode, por exemplo, ser associado com uniformidade, tal como uma medida de variação. Uma estimativa do valor RD é frequentemente benéfica de utilizar, porque ela diminui uma quantidade de processamento requerida quando determinando a qualidade de codificação. O valor da contagem de bits usado
20 no cálculo do valor RD é frequentemente estimado, porque o valor atual é frequentemente difícil de calcular, uma vez que depende de outros valores de dados, e do método de codificação de entropia selecionado. Os bits podem ser estimados baseados na quantidade de bits necessários para liberação de valor, multiplicados pelo fator de compressão estimado que possa ser
25 pré-definido para o fluxo de dados ou podem ser modificados de forma adaptável baseados nos dados comprimidos e liberados anteriores.

Quando se implementa as modalidades mencionadas acima, a compressão de dados utilizada inclui duas fases distintas. Beneficamente, um método para compressão de dados é selecionado, que produz informação
30 de seleção, em que a informação de seleção define o método e os tipos de dados quantificados gerados pelo método. Beneficamente, a informação de seleção é codificada por entropia, por exemplo, para inclusão nos dados

de saída 30. Quando gerando os dados de saída 30, também é opcionalmente benéfico utilizar múltiplos fluxos, por exemplo, para imagens totais ou para blocos Init. O uso de tais múltiplos fluxos é benéfico na redução de entropia nos dados de saída comprimidos 30.

5 As modificações para as modalidades da invenção descrita acima são possíveis sem sair do escopo da invenção como definido pelas reivindicações anexas. Expressões tais como "incluindo", "compreendendo", "incorporando", "consistindo de", "ter", "é" usadas para descrever e reivindicar a presente invenção são destinadas para serem interpretadas de um
10 modo não exclusivo, a saber, permitindo que itens, componentes ou elementos não explicitamente descritos também estejam presentes. Referência ao singular também deve ser interpretada para se referir ao plural. Os números incluídos entre parênteses nas reivindicações anexas são destinados a auxiliar o entendimento das reivindicações e não devem ser interpretados de
15 modo algum para limitar a matéria objeto reivindicada por estas reivindicações.

REIVINDICAÇÕES

1. Método de codificar dados de entrada (20) para gerar dados de saída codificados (30) correspondentes em um codec, **caracterizado** pelo fato de que o método inclui as etapas de:

(a) dividir os dados de entrada (20) em uma pluralidade de blocos ou pacotes, os blocos ou pacotes tendo um tamanho dependendo de uma natureza de seus conteúdos de dados, e/ou tipos de dados e os blocos ou pacotes que são de um ou mais tamanhos;

(b) aplicar uma pluralidade de diferentes tipos de transformações ao conteúdo dos blocos ou pacotes para gerar dados transformados correspondentes;

(c) verificar uma qualidade de representação dos dados transformados dos blocos ou pacotes comparados ao conteúdo dos blocos ou pacotes antes da aplicação da pluralidade de diferentes tipos de transformações para determinar se ou não a qualidade de representação dos dados transformados satisfaz um ou mais critérios de qualidade;

(d) em um evento em que a qualidade de representação dos dados transformados do um ou mais blocos ou pacotes não satisfaz o um ou mais critérios de qualidade, subdividir e/ou combinar ainda o um ou mais blocos ou pacotes individuais cuja qualidade da representação não satisfaz o um ou mais critérios e repetir a etapa (b) para o um ou mais blocos ou pacotes cuja qualidade de representação não satisfaz o um ou mais critérios; e

(e) em um evento em que a qualidade de representação dos dados transformados do um ou mais blocos ou pacotes satisfaz o um ou mais critérios de qualidade, produzindo os dados transformados para prover os dados de saída codificados (30) representativos dos dados de entrada (20) a serem codificados.

2. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que o método inclui pelo menos um de: o uso de uma pluralidade de diferentes tipos de transformações para comprimir o conteúdo associado com os blocos ou pacotes, de modo que os dados de saída codificados são menores em tamanho do que os dados de entrada a serem codificados.

3. Método, de acordo com a reivindicação 1 ou 2, **caracterizado** pelo fato de que o método inclui dispor o um ou mais critérios de qualidade para serem dinamicamente variáveis durante a codificação dos blocos ou pacotes dependendo de uma natureza do conteúdo incluído dentro dos blocos ou pacotes.

4. Método, de acordo com a reivindicação 1, 2 ou 3, **caracterizado** pelo fato de que os blocos ou pacotes são subdivididos e/ou combinados de modo que pelo menos um de seus parâmetros representativos descrevendo seu conteúdo é substancialmente liso dentro de seus blocos ou pacotes subdivididos e/ou combinados.

5. Método, de acordo com qualquer uma das reivindicações 1 a 4, **caracterizado** pelo fato de que os blocos ou pacotes correspondem a uma série de dados, em que a subdivisão dos dados de entrada correspondendo a uma determinada informação para formar a pluralidade de blocos correspondentes é feita dependendo do conteúdo presente em um ou mais dados que precedem a determinada informação dentro da sequência temporal de dados.

6. Método, de acordo com qualquer uma das reivindicações 1 a 5, **caracterizado** pelo fato de que o método inclui adicionar informação de cabeçalho aos dados transformados na etapa (e) para gerar os dados de saída codificados, em que a informação de cabeçalho inclui informação indicativa das transformações empregadas na etapa (b).

7. Método, de acordo com qualquer uma das reivindicações precedentes, **caracterizado** pelo fato de que a etapa (b) inclui buscar informação suplementar a partir de um arranjo de base de dados para uso quando se executa dita pluralidade de diferentes tipos de transformações, em que a informação suplementar inclui pelo menos um de: algoritmos, regras, um ou mais parâmetros de transformação.

8. Método, de acordo com a reivindicação 7, **caracterizado** pelo fato de que o método inclui ainda adicionar informação de cabeçalho aos dados de saída codificados indicativa do arranjo de base de dados para

possibilitar a decodificação subsequente dos dados de saída codificados para acessar dita informação suplementar quando se decodifica os dados de saída codificados.

9. Método, de acordo com qualquer uma das reivindicações precedentes, **caracterizado** pelo fato de que o método inclui utilizar para a pluralidade de diferentes tipos de transformações um de mais dos seguintes métodos de transformação usados para a codificação: uma referência de base de dados, um valor DC, método de codificação de slides, método de codificação de escala, método de codificação de linha, multinível, inalterada, método de codificação de interpolação, método de codificação de extrapolação, transformada discreta de cosseno (DCT), modulação de por código de pulso (PCM), modulação por codificação diferencial de pulso (DPCM), RLE, SRLE, EM, LZO, VLC, codificação de Huffman, codificação aritmética, codificação de faixa, codifiicação de transformação, codificação delta, codificação ODelta, RLE específico de bzip2.

10. Método, de acordo com qualquer uma das reivindicações precedentes, **caracterizado** pelo fato de que o método inclui codificar pelo menos um de vídeo, imagem, áudio, gráficos, dados econômicos, dados de máscara, dados multidimensionais (3D), dados de texto, ECG, sísmico, ASCII, Unicode, dados de calendário, dados matemáticos e informação binária presentes nos dados de entrada.

11. Método, de acordo com qualquer uma das reivindicações precedentes, **caracterizado** pelo fato de que o método inclui codificar múltiplos canais e/ou camadas nos dados de saída codificados para prover pelo menos um de: vídeo interativo, anúncios comerciais, uma pluralidade de pontos de vista durante reportagem de evento esportivo, marcas d'água interativas, reconhecimento de padrão interativo, e botões de interface de usuário 2D/3D animados.

12. Método, de acordo com qualquer uma das reivindicações precedentes, **caracterizado** pelo fato de que o método inclui prover os dados codificados a pelo menos uma das seguintes destinações: um dispositivo de memória de dados, uma rede de comunicação, um cartão de memória, discos

de memória de dados, redes de comunicação de área local (LANs), diretamente a um decodificador.

13. Método, de acordo com qualquer uma das reivindicações precedentes, **caracterizado** pelo fato de que a etapa (a) do método inclui inicialmente dividir os dados de entrada (20) em um ou mais blocos baseada em pelo menos um de:

- (a) resoluções de imagem;
- (b) uma quantidade de dados;
- (c) um conteúdo de dados;
- (c d) tamanho de bloco ou pacote máximo; e
- (e) um parâmetro de qualidade.

14. Uma mídia de armazenamento não transitória legível por computador que compreende instruções de computador, as quais, quando executadas por um processador, fazem com que o processador execute o método conforme reivindicado na reivindicação 1.

15. Codificador (10) operável para codificar dados de entrada (20) para gerar dados de saída codificados (30) correspondentes, **caracterizado** pelo fato de que o codificador (10) inclui hardware de processamento de dados que é operável:

(a) para dividir os dados de entrada (20) em uma pluralidade de blocos ou pacotes, os blocos ou pacotes tendo um tamanho dependendo de uma natureza de seus conteúdos de dados, e/ou tipos de dados e os blocos ou pacotes que são de um ou mais tamanhos;

(b) para aplicar uma pluralidade de diferentes tipos de transformações ao conteúdo dos blocos ou pacotes para gerar dados transformados correspondentes;

(c) para verificar uma qualidade de representação dos dados transformados dos blocos ou pacotes comparados ao conteúdo dos blocos ou pacotes antes da aplicação da pluralidade de diferentes tipos de transformações para determinar se ou não a qualidade de representação dos dados transformados satisfaz um ou mais critérios de qualidade;

(d) em um evento em que a qualidade de representação dos dados transformados do um ou mais blocos ou pacotes não satisfaz o um ou mais critérios de qualidade, subdividir e/ou combinar ainda o um ou mais blocos ou pacotes individuais cuja qualidade da representação não satisfaz o um ou mais critérios e repetir a etapa (b) para o um ou mais blocos ou pacotes cuja qualidade de representação não satisfaz o um ou mais critérios; e

(e) em um evento em que a qualidade de representação dos dados transformados do um ou mais blocos ou pacotes satisfaz o um ou mais critérios de qualidade, produzindo os dados transformados para prover os dados de saída codificados (30) representativos dos dados de entrada (20) a serem codificados.

16. Codificador (10), de acordo com a reivindicação 15, **caracterizado** pelo fato de que o codificador (10) é operável para usar a pluralidade de diferentes tipos de transformações para comprimir o conteúdo associado com os blocos ou pacotes, de modo que os dados de saída (30) codificados são menores em tamanho do que os dados de entrada (20) a serem codificados.

17. Codificador (10), de acordo com a reivindicação 15 ou 16, **caracterizado** pelo fato de que o codificador (10) é operável para dispor o um ou mais critérios de qualidade para ser dinamicamente variável durante a codificação dos blocos ou pacotes dependendo de uma natureza do conteúdo incluído dentro dos blocos ou pacotes.

18. Codificador (10), de acordo com qualquer uma das reivindicações 15 a 17, **caracterizado** pelo fato de que os blocos ou pacotes são subdivididos e/ou combinados de modo que pelo menos um de seus parâmetros representativos descrevendo seu conteúdo é substancialmente liso dentro de seus blocos ou pacotes subdivididos e/ou combinados.

19. Codificador (10), de acordo com qualquer uma das reivindicações 15 a 18, **caracterizado** pelo fato de que os blocos ou pacotes correspondem a uma série de dados, em que a subdivisão dos dados de entrada (20) correspondendo a uma determinada informação para formar uma pluralidade de blocos correspondentes torna-se dependente do conteúdo presente em um ou mais dados que precedem a determinada informação dentro da sequência temporal de dados.

20. Codificador (10), de acordo com qualquer uma das reivindicações 15 a 19, **caracterizado** pelo fato de que o codificador (10) é operável para adicionar informação de cabeçalho aos dados transformados para gerar os dados de entrada codificados (30), em que a informação de cabeçalho inclui informação indicativa das transformações empregadas pelo codificador (10).

21. Codificador (10), de acordo com qualquer uma das reivindicações 15 a 20, **caracterizado** pelo fato de que o codificador (10) é operável para buscar informação suplementar a partir de um arranjo de base de dados para uso quando se executa as transformações, em que a informação suplementar inclui pelo menos um dos algoritmos, regras, um ou mais parâmetros de transformação.

22. Codificador (10), de acordo com a reivindicação 21, **caracterizado** pelo fato de que o codificador (10) é operável para adicionar informação de cabeçalho aos dados de saída codificados em um modo indicativo do arranjo de base de dados para possibilitar a decodificação subsequente dos dados de saída codificados para acessar a informação suplementar quando se decodifica os dados de saída codificados.

23. Codificador (10), de acordo com qualquer uma das reivindicações 15 a 22, **caracterizado** pelo fato de que o codificador (10) é operável para utilizar a pluralidade de diferentes tipos de as transformações de um ou mais de: uma referência de base de dados, um valor DC, método de codificação de slides, método de codificação de escala, método de codificação de linha, multinível, inalterada, método de codificação de interpolação, método de codificação de extrapolação, transformada discreta de cosseno (DCT), modulação por código de pulso (PCM), modulação por codificação diferencial de pulso (DPCM), RLE, SRLE, EM, LZO, VLC, codificação de Huffman, codificação aritmética, codificação de faixa, codificação de transformação, codificação delta, codificação ODelta, RLE específico de bzip2.

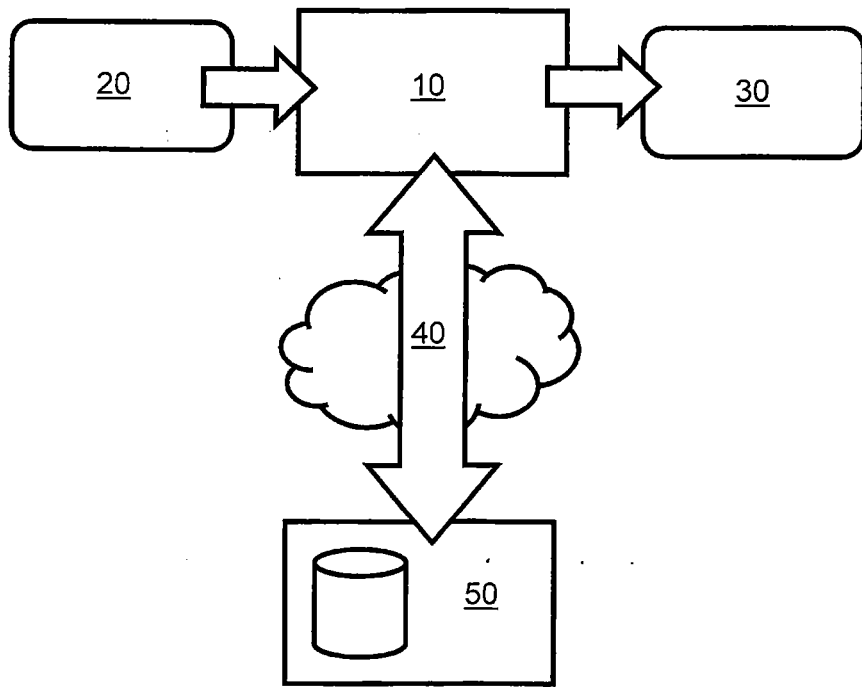
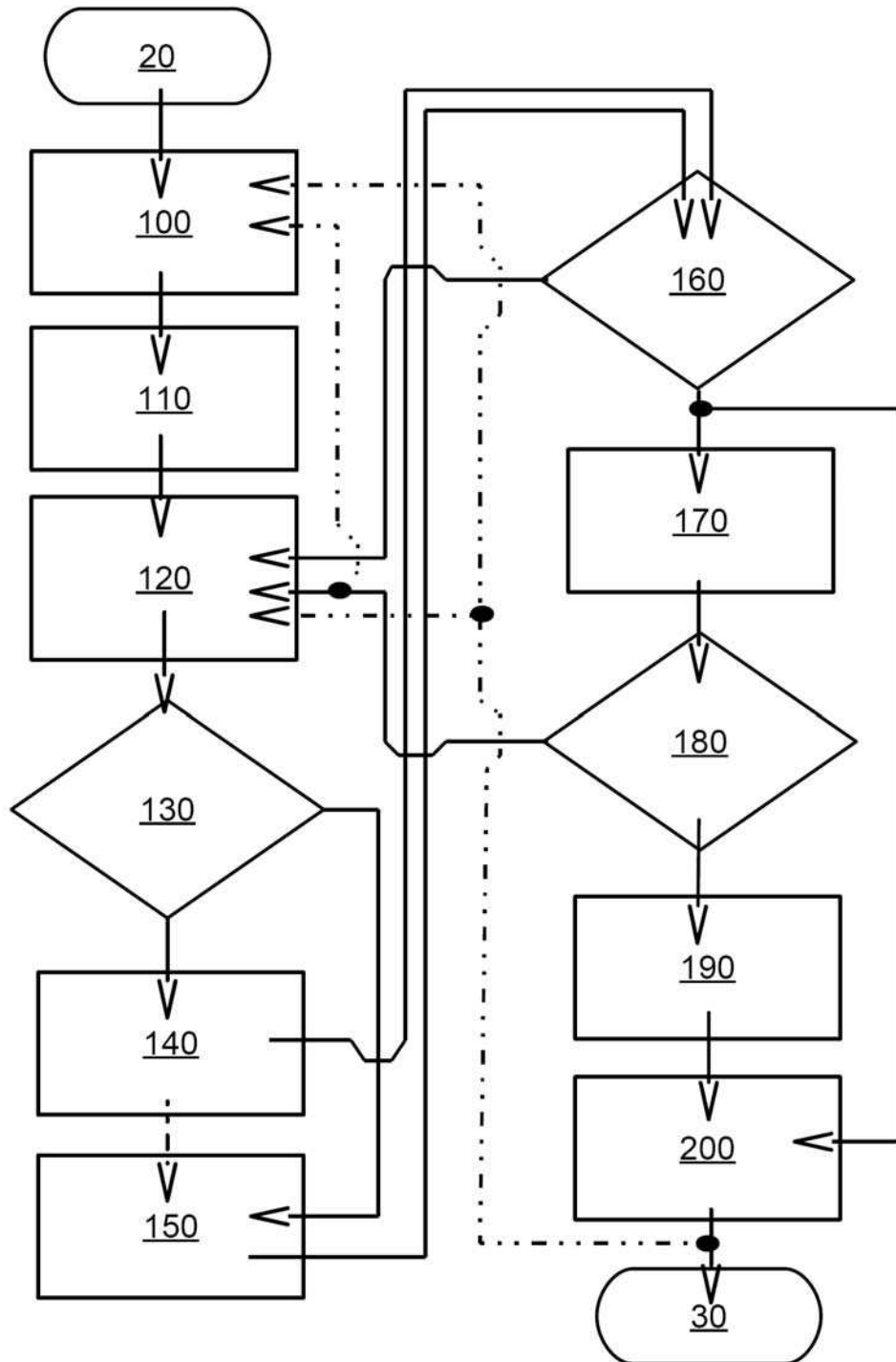


FIG. 1

**FIG.2**

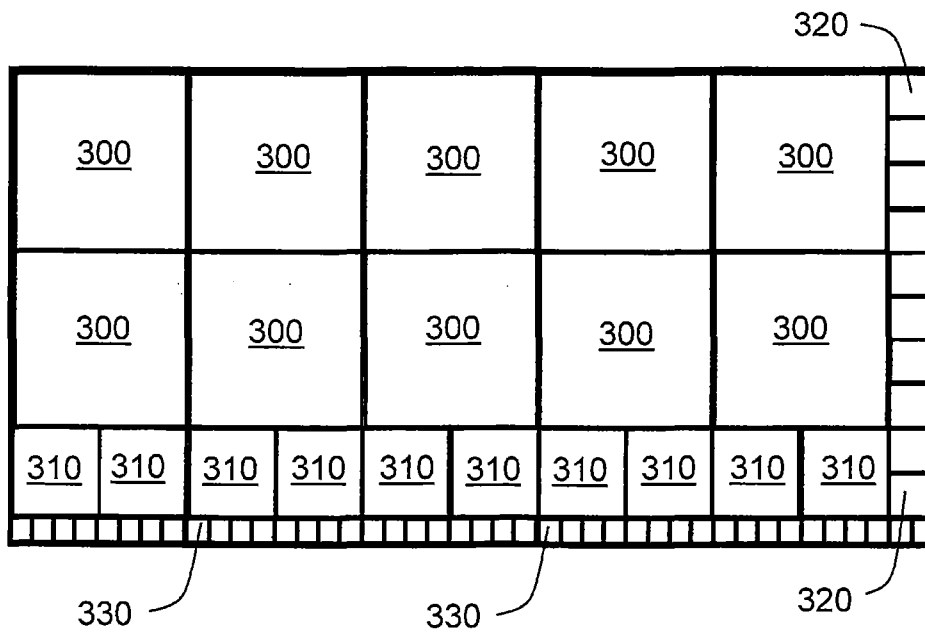


FIG. 3

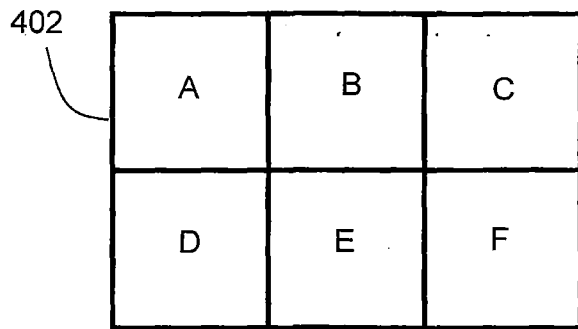
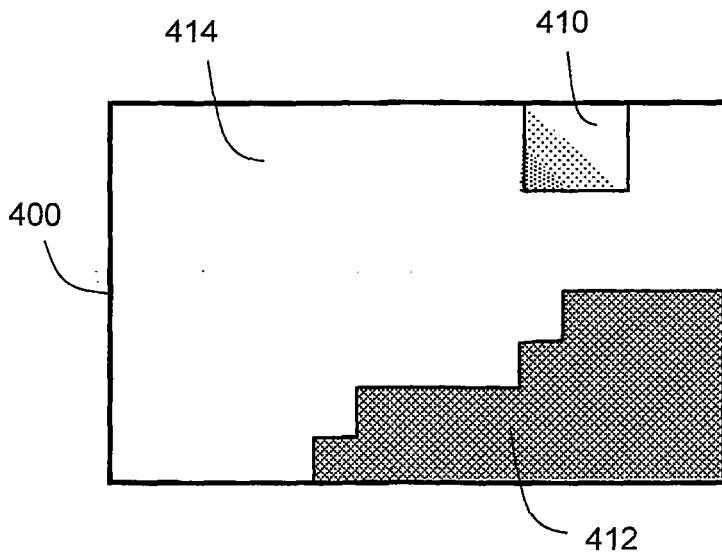


FIG. 4

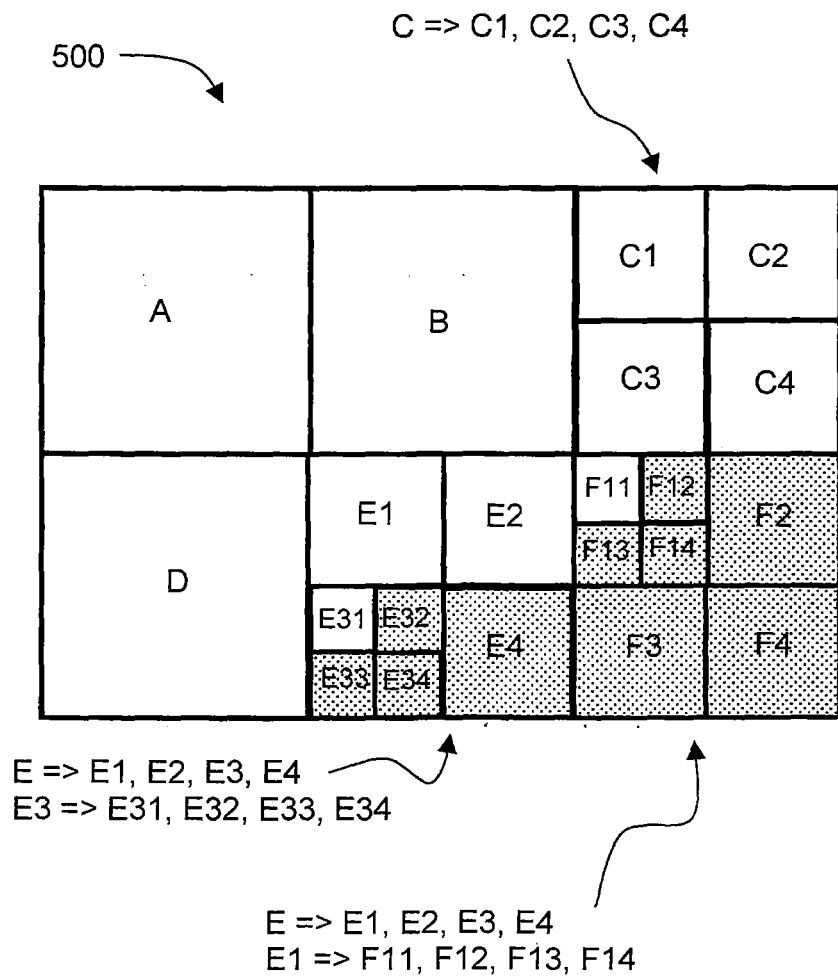


FIG. 5

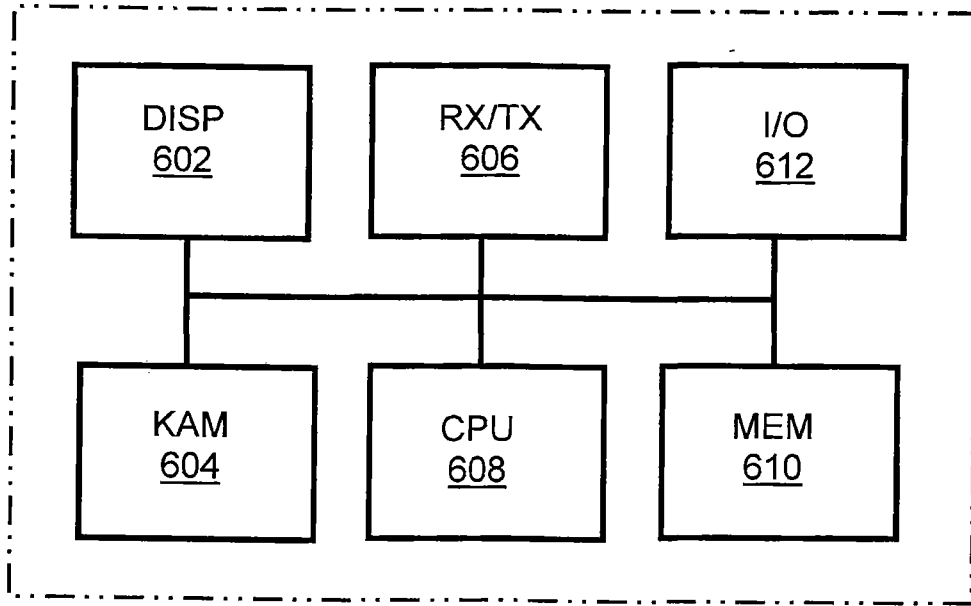


FIG. 6