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

(21) **BR 12 2012 021799-9 A2**

* B R 1 2 2 0 1 2 0 2 1 7 9 9 A 2 *

(22) Data de Depósito: 16/09/2008

(43) Data da Publicação: 04/08/2015
(RPI 2326)

(54) Título: MÉTODO PARA INCORPORAR INFORMAÇÃO DE USABILIDADE DE VÍDEO (VUI) EM UM SISTEMA DE CODIFICAÇÃO DE VÍDEO DE MÚLTIPLAS VISUALIZAÇÕES (MVC)

(51) Int.Cl.: H04N19/46

(52) CPC: H04N19/46

(30) Prioridade Unionista: 05/10/2007 US 60/977,709

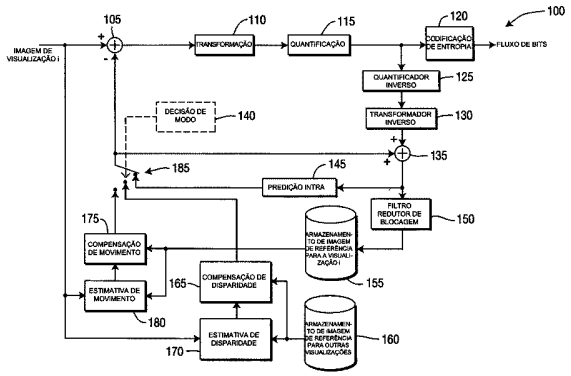
(73) Titular(es): Thomson Licensing

(72) Inventor(es): Jiancong Luo, Peng Yin

(86) Pedido Internacional: PCT US2008010775 de 16/09/2008

(87) Publicação Internacional: WO 2009/048502 de 16/04/2009

(57) Resumo: MÉTODO PARA INCORPORAR INFORMAÇÃO DE USABILIDADE DE VÍDEO (VUI) EM UM SISTEMA DE CODIFICAÇÃO DE VÍDEO DE MÚLTIPLAS VISUALIZAÇÕES (MVC). São fornecidos métodos e aparelho para incorporar Informação de Usabilidade de Vídeo (VUI) em condificação de vídeo de múltiplas visualizações (MVC). Um aparelho (100) inclui um codificador (100) para codificar conteúdo de vídeo de mútiplas visualizações ao especificar Informação de Usabilidade de Vídeo para pelo menos um selecionado de: visualizações individuis (300), níveis temporais individuais em uma visualização (500) e pontos de operação individuais (700). Adicionalmente, um aparelho (200) inclui um decodificador para decodificar conteúdo de vídeo de múltiplas visualizações ao especificar Informação de Usabilidade de Vídeo para pelo menos um selecionado de: visualizações individuais (400), níveis temporais individuais em uma visualização (600) e pontos de operação individuais (800).



"MÉTODO PARA INCORPORAR INFORMAÇÃO DE USABILIDADE DE VÍDEO (VUI) EM UM SISTEMA DE CODIFICAÇÃO DE VÍDEO DE MÚLTIPLAS VISUALIZAÇÕES (MVC)" – Dividido do pedido PI0817420-2, depositado em 16 de setembro de 2008

REFERÊNCIA CRUZADA A PEDIDOS RELACIONADOS

Este pedido reivindica o benefício do pedido provisório US 60/977.709, depositado em 5 de outubro de 2007, o qual está incorporado na sua totalidade neste documento pela referência. Adicionalmente, este pedido está relacionado ao pedido não provisório de protocolo representante No. PU070239, intitulado "METHODS AND APPARATUS FOR INCORPORATING VIDEO USABILITY INFORMATION (VUI) WITHIN A MULTI-VIEW VIDEO (MVC) CODING SYSTEM", o qual também reivindica o benefício do pedido provisório US 60/977.709, depositado em 5 de outubro de 2007, e que é do mesmo requerente, incorporado neste documento pela referência e depositado concorrentemente com este.

CAMPO TÉCNICO

Os presentes princípios dizem respeito de uma maneira geral a codificação e decodificação de vídeo e, mais particularmente, a métodos e aparelho para incorporar informação de usabilidade de vídeo (VUI) em codificação de vídeo de múltiplas visualizações (MVC).

ANTECEDENTES

A Organização Internacional de Normalização/Comissão Internacional de Eletrotécnica (ISO/IEC) Grupo-4 de Especialistas de Imagens em Movimento (MPEG-4), padrão de Codificação Avançada de Vídeo (AVC) Parte 10/União Internacional de Telecomunicações, Setor de Telecomunicações (ITU-T) recomendação H.264 (em seguida o "padrão MPEG-4 AVC") especifica sintaxe e semânticas de parâmetros de informação de usabilidade de vídeo (VUI) de conjuntos de parâmetros de sequência. Informação de usabilidade de vídeo inclui informação de relação de aspectos, sobrevarredura, tipo de sinal de vídeo, localização de croma, sincronismo, parâmetros de decodificador hipotético de referência (HRD) de camada de abstração de rede (NAL), parâmetros de decodificador hipotético de referência de camada de codificação de vídeo (VCL), restrição de fluxo de bits e assim por diante. Informação de usabilidade de vídeo fornece informação extra para um fluxo de bits correspondente para permitir uma aplicação mais ampla para um usuário. Por exemplo, em informação de restrição de fluxo de bits, informação de usabilidade de vídeo especifica: (1) se o movimento está acima de um limite de imagem; (2) os bytes máximos por imagem; (3) os bits máximos por macrobloco; (4) o comprimento de vetor de movimento máximo (horizontal e vertical); (5) o número de quadros de reordenação; e (6) o tamanho de armazenamento temporário de quadro decodificado máximo. Quando o decodificador vê a informação, em vez de usar a informação de "nível" para estabelecer a exigência de decodificação, a qual em geral é maior que aquela que o fluxo de bits realmente exige, o decodificador pode cus-

tomizar sua operação de decodificação com base no limite mais apertado.

Codificação de vídeo de múltiplas visualizações (MVC) é uma extensão para o Padrão MPEG-4 AVC. Em codificação de vídeo de múltiplas visualizações, imagens de vídeo para múltiplas visualizações podem ser codificadas por meio de explorar a correlação entre visualizações. Entre todas as visualizações, uma visualização é a visualização de base, a qual é compatível com Padrão MPEG-4 AVC e não pode ser predita a partir das outras visualizações. As outras visualizações são referidas como visualizações não de base. Visualizações não de base podem ser codificadas preditivamente a partir da visualização de base e de outras visualizações não de base. Cada visualização pode ser subamostrada temporalmente. Um subconjunto temporal de uma visualização pode ser identificado por meio de um elemento de sintaxe temporal_id. Um nível temporal de uma visualização é uma representação do sinal de vídeo. Podem existir diferentes combinações de visualizações e níveis temporais em um fluxo de bits codificado de vídeo de múltiplas visualizações. Cada combinação é chamada de ponto de operação. Subfluxos de bits correspondendo aos pontos de operação podem ser extraídos do fluxo de bits.

SUMÁRIO

Estes e outros inconvenientes e desvantagens da técnica anterior são abordados pelos presentes princípios, os quais são direcionados para métodos e aparelho para incorporar informação de usabilidade de vídeo (VUI) em codificação de vídeo de múltiplas visualizações (MVC).

De acordo com um aspecto dos presentes princípios, é fornecido um aparelho. O aparelho inclui um codificador para codificar conteúdo de vídeo de múltiplas visualizações ao especificar informação de usabilidade de vídeo para pelo menos um de visualizações individuais, níveis temporais individuais em uma visualização e pontos de operação individuais.

De acordo com um outro aspecto dos presentes princípios, é fornecido um método. O método inclui codificar conteúdo de vídeo de múltiplas visualizações ao especificar informação de usabilidade de vídeo para pelo menos um de visualizações individuais, níveis temporais individuais em uma visualização e pontos de operação individuais.

De acordo também com um outro aspecto dos presentes princípios, é fornecido um aparelho. O aparelho inclui um decodificador para decodificar conteúdo de vídeo de múltiplas visualizações ao especificar informação de usabilidade de vídeo para pelo menos um de visualizações individuais, níveis temporais individuais em uma visualização e pontos de operação individuais.

De acordo com ainda um outro aspecto dos presentes princípios, é fornecido um método. O método inclui decodificar conteúdo de vídeo de múltiplas visualizações ao especificar informação de usabilidade de vídeo para pelo menos um de visualizações individuais,

níveis temporais individuais em uma visualização e pontos de operação individuais.

Estes e outros aspectos, recursos e vantagens dos presentes princípios se tornarão aparentes a partir da descrição detalhada a seguir de modalidades exemplares, a qual é para ser lida em conexão com os desenhos anexos.

DESCRIÇÃO RESUMIDA DOS DESENHOS

Os presentes princípios podem ser mais bem entendidos de acordo com as figuras exemplares seguintes, nas quais:

A figura 1 é um diagrama de blocos para um codificador de Codificação de Vídeo de Múltiplas Visualizações (MVC) exemplar ao qual os presentes princípios podem ser aplicados, de acordo com uma modalidade dos presentes princípios;

A figura 2 é um diagrama de blocos para um decodificador de Codificação de Vídeo de Múltiplas Visualizações (MVC) exemplar ao qual os presentes princípios podem ser aplicados, de acordo com uma modalidade dos presentes princípios;

A figura 3 é um fluxograma para um método exemplar para codificar parâmetros de restrição de fluxo de bits para cada visualização, usando um elemento de sintaxe mvc_vui_parameters_extension(), de acordo com uma modalidade dos presentes princípios;

A figura 4 é um fluxograma para um método exemplar para decodificar parâmetros de restrição de fluxo de bits para cada visualização, usando um elemento de sintaxe mvc_vui_parameters_extension(), de acordo com uma modalidade dos presentes princípios;

A figura 5 é um fluxograma para um método exemplar para codificar parâmetros de restrição de fluxo de bits para cada nível temporal em cada visualização, usando um elemento de sintaxe mvc_vui_parameters_extension(), de acordo com uma modalidade dos presentes princípios;

A figura 6 é um fluxograma para um método exemplar para decodificar parâmetros de restrição de fluxo de bits para cada nível temporal em cada visualização, usando um elemento de sintaxe mvc_vui_parameters_extension(), de acordo com uma modalidade dos presentes princípios;

A figura 7 é um fluxograma para um método exemplar para codificar parâmetros de restrição de fluxo de bits para cada ponto de operação, usando um elemento de sintaxe view_scalability_parameters_extension(), de acordo com uma modalidade dos presentes princípios; e

A figura 8 é um fluxograma para um método exemplar para decodificar parâmetros de restrição de fluxo de bits para cada ponto de operação, usando um elemento de sintaxe view_scalability_parameters_extension(), de acordo com uma modalidade dos presentes princípios.

DESCRIÇÃO DETALHADA

Os presentes princípios são direcionados para métodos e aparelho para incorporar

informação de usabilidade de vídeo (VUI) em codificação de vídeo de múltiplas visualizações (MVC).

A presente descrição ilustra os presentes princípios. Assim, ficará reconhecido que os versados na técnica serão capazes de imaginar vários arranjos que, embora não descritos ou mostrados explicitamente neste documento, incorporam os presentes princípios e estão incluídos no seu espírito e escopo.

Todos os exemplos e linguagem condicional relatados neste documento são pretendidos para propósitos pedagógicos para ajudar o leitor a entender os presentes princípios e os conceitos contribuídos pelo(s) inventor(s) para incrementar a técnica, e são para serem interpretados como sendo sem limitação para tais exemplos e condições especificamente relatados.

Além disso, todas as declarações neste documento relacionando princípios, aspectos e modalidades dos presentes princípios, assim como exemplos específicos dos mesmos, são pretendidas para abranger tanto equivalências estruturais quanto funcionais dos mesmos. Adicionalmente, é considerado que tais equivalências incluem tanto equivalências atualmente conhecidas quanto equivalências desenvolvidas no futuro, isto é, quaisquer elementos desenvolvidos que executem a mesma função, independente de estrutura.

Assim, por exemplo, será percebido pelos versados na técnica que os diagramas de blocos apresentados neste documento representam visualizações conceituais de conjunto de circuitos ilustrativos incorporando os presentes princípios. De forma similar, será percebido que quaisquer fluxogramas, diagramas, diagramas de transição de estado, pseudocódigo e outros mais representam vários processos que podem ser substancialmente representados em mídias legíveis por computador e assim executados por um computador ou processador, se tal computador ou processador está ou não mostrado explicitamente.

As funções dos vários elementos mostrados nas figuras podem ser fornecidas por meio do uso de hardware dedicado assim como de hardware capaz de executar software em associação com software apropriado. Quando fornecidas por um processador, as funções podem ser fornecidas por um único processador dedicado, por um único processador compartilhado, ou por uma pluralidade de processadores individuais, alguns dos quais podem ser compartilhados. Além disso, uso explícito do termo "processador" ou "controlador" não deve ser interpretado para se referir exclusivamente a hardware capaz de executar software, e implicitamente pode incluir, sem limitação, hardware de processador de sinal digital ("DSP"), memória somente de leitura ("ROM") para armazenar software, memória de acesso aleatório ("RAM") e armazenamento não volátil.

Outro hardware, convencional e/ou feito sob medida, também pode ser incluído. De forma similar, quaisquer comutadores mostrados nas figuras são somente conceituais. Sua função pode ser executada por meio da operação de lógica de programa, por meio de lógica

dedicada, por meio da interação de controle de programa e lógica dedicada, ou mesmo manualmente, a técnica particular sendo selecionável pelo implementador tal com entendido mais especificamente a partir do contexto.

Nas reivindicações deste pedido, qualquer elemento expressado como um dispositivo para executar uma função especificada é pretendido para abranger algum modo de executar essa função incluindo, por exemplo, a) uma combinação de elementos de circuito que executa essa função ou b) software em qualquer forma, incluindo, portanto, firmware, microcódigo ou coisa parecida, combinado com conjunto de circuitos apropriados para executar esse software para executar a função. Os presentes princípios, tal como definido pelas tais reivindicações, residem no fato de que as funcionalidades fornecidas pelos vários dispositivos relatados são combinadas e levadas conjuntamente no modo que as reivindicações requerem. Assim, é considerado que quaisquer dispositivos que possam fornecer essas funcionalidades são equivalentes a esses mostrados neste documento.

Referência na especificação a "uma modalidade" dos presentes princípios significa que um recurso, estrutura, característica particular e assim por diante descrito em conexão com a modalidade está incluído em pelo menos uma modalidade dos presentes princípios. Assim, os significados da frase "em uma modalidade" aparecendo em vários lugares por toda a especificação não estão todos necessariamente se referindo à mesma modalidade.

É para ser percebido que o uso das expressões "e/ou" e "pelo menos um de", por exemplo, nos casos de "A e/ou B" e "pelo menos um de A e B", é pretendido para abranger somente a seleção da primeira opção listada (A), ou somente a seleção da segunda opção listada (B), ou a seleção de ambas as opções (A e B). Como um exemplo adicional, nos casos de "A, B e/ou C" e "pelo menos um de A, B e C", tal fraseado é pretendido para abranger somente a seleção da primeira opção listada (A), ou somente a seleção da segunda opção listada (B), ou somente a seleção da terceira opção listada (C), ou somente a seleção das primeira e segunda opções listadas (A e B), ou somente a seleção das primeira e terceira opções listadas (A e C), ou somente a seleção das segunda e terceira opções listadas (B e C), ou a seleção de todas as três opções (A e B e C). Isto pode ser estendido, tal como prontamente aparente para uma pessoa de conhecimento comum nesta técnica e em técnicas relacionadas, para tantos itens quanto listados.

Codificação de Vídeo de Múltiplas Visualizações (MVC) é a estrutura de compressão para a codificação de sequências de múltiplas visualizações. Uma sequência de codificação de vídeo de múltiplas visualizações (MVC) é um conjunto de duas ou mais sequências de vídeo que capturam a mesma cena a partir de um diferente ponto de visualização.

Tal como usado de forma intercambiável neste documento, "visualização cruzada" e "intervisualização" se referem a imagens que pertencem a uma visualização a não ser uma visualização atual.

Além disso, tal como usado neste documento, "sintaxe de alto nível" se refere à sintaxe presente no fluxo de bits que reside hierarquicamente acima da camada de macrobloco. Por exemplo, sintaxe de alto nível, tal como usado neste documento, pode se referir a, mas não está limitada a isto, sintaxe no nível de cabeçalho de fatia, nível de Informação Suplementar de Enriquecimento (SEI), nível de Conjunto de Parâmetros de Imagem (PPS), nível de Conjunto de Parâmetros de Sequência (SPS) e nível de cabeçalho de unidade de Camada de Abstração de Rede (NAL).

Também, é para ser percebido que embora uma ou mais modalidades dos presentes princípios estejam descritas neste documento para propósitos ilustrativos com relação à extensão de codificação de vídeo de múltiplas visualizações do padrão MPEG-4 AVC, os presentes princípios não estão limitados unicamente a esta extensão e/ou a este padrão e, assim, podem ser utilizados com relação a outros padrões, recomendações e extensões de codificação de vídeo dos mesmos, enquanto mantendo o espírito dos presentes princípios.

Adicionalmente, é para ser percebido que embora uma ou mais modalidades dos presentes princípios estejam descritas neste documento para propósitos ilustrativos com relação a informação de restrição de fluxo de bits, os presentes princípios não estão limitados a usar unicamente informação de restrição de fluxo de bits como um tipo de informação de usabilidade de vídeo e, assim, outros tipos de informação de usabilidade de vídeo que podem ser estendidos para uso com relação a codificação de vídeo de múltiplas visualizações também pode ser usados de acordo com os presentes princípios, enquanto mantendo o espírito dos presentes princípios.

Voltando à figura 1, um codificador de Codificação de Vídeo de Múltiplas Visualizações (MVC) exemplar está indicado de uma maneira geral pelo número de referência 100. O codificador 100 inclui um combinador 105 tendo uma saída conectada em comunicação de sinal com uma entrada de um transformador 110. Uma saída do transformador 110 é conectada em comunicação de sinal com uma entrada do quantificador 115. Uma saída do quantificador 115 é conectada em comunicação de sinal com uma entrada de um codificador de entropia 120 e com uma entrada de um quantificador inverso 125. Uma saída do quantificador inverso 125 é conectada em comunicação de sinal com uma entrada de um transformador inverso 130. Uma saída do transformador inverso 130 é conectada em comunicação de sinal com uma primeira entrada não inversora de um combinador 135. Uma saída do combinador 135 é conectada em comunicação de sinal com uma entrada de um preditor intra 145 e com uma entrada de um filtro redutor de blocagem 150. Uma saída do filtro redutor de blocagem 150 é conectada em comunicação de sinal com uma entrada de um armazenamento de imagem de referência 155 (para a visualização i). Uma saída do armazenamento de imagem de referência 155 é conectada em comunicação de sinal com uma primeira entrada de um compensador de movimento 175 e com uma primeira entrada de um estimador de mo-

vimento 180. Uma saída do estimador de movimento 180 é conectada em comunicação de sinal com uma segunda entrada do compensador de movimento 175.

Uma saída de um armazenamento de imagem de referência 160 (para outras visualizações) é conectada em comunicação de sinal com uma primeira entrada de um estimador de disparidade/iluminação 170 e com uma primeira entrada de um compensador de disparidade/iluminação 165. Uma saída do estimador de disparidade/iluminação 170 é conectada em comunicação de sinal com uma segunda entrada do compensador de disparidade/iluminação 165.

Uma saída do decodificador de entropia 120 está disponível como uma saída do codificador 100. Uma entrada não inversora do combinador 105 está disponível como uma entrada do codificador 100, e é conectada em comunicação de sinal com uma segunda entrada do estimador de disparidade/iluminação 170 e com uma segunda entrada do estimador de movimento 180. Uma saída de um comutador 185 é conectada em comunicação de sinal com uma segunda entrada não inversora do combinador 135 e com uma entrada inversora do combinador 105. O comutador 185 inclui uma primeira entrada conectada em comunicação de sinal com uma saída do compensador de movimento 175, uma segunda entrada conectada em comunicação de sinal com uma saída do compensador de disparidade/iluminação 165 e uma terceira entrada conectada em comunicação de sinal com uma saída do preditor intra 145.

Um módulo de decisão de modo 140 tem uma saída conectada ao comutador 185 para controlar qual entrada é selecionada pelo comutador 185.

Voltando à figura 2, um decodificador de Codificação de Vídeo de Múltiplas Visualizações (MVC) exemplar está indicado de uma maneira geral pelo número de referência 200. O decodificador 200 inclui um decodificador de entropia 205 tendo uma saída conectada em comunicação de sinal com uma entrada de um quantificador inverso 210. Uma saída do quantificador inverso é conectada em comunicação de sinal com uma entrada de um transformador inverso 215. Uma saída do transformador inverso 215 é conectada em comunicação de sinal com uma primeira entrada não inversora de um combinador 220. Uma saída do combinador 220 é conectada em comunicação de sinal com uma entrada de um filtro redutor de blocagem 225 e com uma entrada de um preditor intra 230. Uma saída do filtro redutor de blocagem 225 é conectada em comunicação de sinal com uma entrada de um armazenamento de imagem de referência 240 (para a visualização i). Uma saída do armazenamento de imagem de referência 240 é conectada em comunicação de sinal com uma primeira entrada de um compensador de movimento 235.

Uma saída de um armazenamento de imagem de referência 245 (para outras visualizações) é conectada em comunicação de sinal com uma primeira entrada de um compensador de disparidade/iluminação 250.

Uma entrada do decodificador de entropia 205 está disponível como uma entrada para o decodificador 200, para receber um fluxo de bits de resíduo. Além disso, uma entrada de um módulo de modo 260 também está disponível como uma entrada para o decodificador 200, para receber sintaxe de controle para controlar qual entrada é selecionada pelo comutador 255. Adicionalmente, uma segunda entrada do compensador de movimento 235 está disponível como uma entrada do decodificador 200, para receber vetores de movimento. Também, uma segunda entrada do compensador de disparidade/iluminação 250 está disponível como uma entrada para o decodificador 200, para receber vetores de disparidade e sintaxe de compensação de iluminação.

Uma saída de um comutador 255 é conectada em comunicação de sinal com uma segunda entrada não inversora do combinador 220. Uma primeira entrada do comutador 255 é conectada em comunicação de sinal com uma saída do compensador de disparidade/iluminação 250. Uma segunda entrada do comutador 255 é conectada em comunicação de sinal com uma saída do compensador de movimento 235. Uma terceira entrada do comutador 255 é conectada em comunicação de sinal com uma saída do preditor intra 230. Uma saída do módulo de modo 260 é conectada em comunicação de sinal com o comutador 255 para controlar qual entrada é selecionada pelo comutador 255. Uma saída do filtro redutor de blocagem 225 está disponível como uma saída do decodificador.

No Padrão MPEG-4 AVC, parâmetros de sintaxe e semântica dos conjuntos de parâmetros de sequência são especificados para informação de usabilidade de vídeo (VUI). Isto representa informação adicional que pode ser inserida em um fluxo de bits para aprimorar a usabilidade do vídeo para uma grande variedade de propósitos. Informação de usabilidade de vídeo inclui informação de relação de aspectos, sobrevarredura, tipo de sinal de vídeo, localização de croma, sincronismo, parâmetros de decodificador hipotético de referência (HRD) de camada de abstração de rede (NAL), parâmetros de decodificador hipotético de referência de camada de codificação de vídeo (VCL), restrição de fluxo de bits e assim por diante.

De acordo com uma ou mais modalidades dos presentes princípios, usamos este campo de informação de usabilidade de vídeo existente para novos e diferentes propósitos do que na técnica anterior e, adicionalmente, estendemos seu uso para codificação de vídeo de múltiplas visualizações (MVC). Em nosso esquema de codificação de vídeo de múltiplas visualizações, a informação de usabilidade de vídeo é estendida de maneira que ela pode ser diferente entre, por exemplo, diferentes visualizações, diferentes níveis temporais em uma visualização, ou diferentes pontos de operação. Assim, de acordo com uma modalidade, especificamos informação de usabilidade de vídeo de acordo com um ou mais de, mas não se limitando a isto, o seguinte: especificar a informação de usabilidade de vídeo para visualizações individuais, especificar a informação de usabilidade de vídeo para níveis tem-

porais individuais em uma visualização e especificar a informação de usabilidade de vídeo para pontos de operação individuais separadamente.

No Padrão MPEG-4 AVC, um conjunto que inclua Informação de Usabilidade de Vídeo (VUI) pode ser transmitido em um conjunto de parâmetros de sequência (SPS). De acordo com uma modalidade, estendemos o conceito de Informação de Usabilidade de Vídeo para uso dentro de um contexto de codificação de vídeo de múltiplas visualizações (MVC). Vantajosamente, isto permite que Informação de Usabilidade de Vídeo diferente seja especificada para diferentes visualizações, diferentes níveis temporais em uma visualização, ou para diferentes pontos de operação em codificação de vídeo de múltiplas visualizações. Em uma modalidade, fornecemos uma abordagem inédita ao considerar, modificar e usar informação de restrição de fluxo de bits em Informação de Usabilidade de Vídeo para codificação de vídeo de múltiplas visualizações.

A informação de restrição de fluxo de bits no Padrão MPEG-4 AVC é especificada no elemento de sintaxe vui_parameters() que é uma parte do sequence_parameter_set(). A TABELA 1 ilustra a sintaxe de Padrão MPEG-4 AVC de vui_parameters().

TABELA 1

vui_parameters(){	C	Descritor
aspect_ratio_info_present_flag	0	u(1)
...		
bitstream_restriction_flag	0	u(1)
if(bitstream_restriction_flag){		
motion_vectors_over_pic_boundaries_flag	0	u(1)
max_bytes_per_pic_denom	0	ue(v)
max_bits_per_mb_denom	0	ue(v)
log2_max_mv_length_horizontal	0	ue(v)
log2_max_mv_length_vertical	0	ue(v)
num_reorder_frames	0	ue(v)
max_dec_frame_buffering	0	ue(v)
}		
}		

As semânticas dos elementos de sintaxe de informação de restrição de fluxo de bits são como se segue:

bitstream_restriction_flag igual a 1 especifica que os parâmetros de restrição de fluxo de bits de sequência de vídeo codificado seguinte estão presentes.

bitstream_restriction_flag igual a 0 especifica que os parâmetros de restrição de fluxo de bits de sequência de vídeo codificado seguinte não estão presentes.

motion_vectors_over_pic_boundaries_flag igual a 0 indica que nenhuma amostra fora dos limites de imagem e nenhuma amostra em uma posição de amostra fracionária cujo valor é derivado usando uma ou mais amostras fora dos limites de imagem são usadas para predição intra de qualquer amostra.

motion_vectors_over_pic_boundaries_flag igual a 1 indica que uma ou mais amostras fora dos limites de imagem podem ser usadas em predição intra. Quando o elemento de sintaxe motion_vectors_over_pic_boundaries_flag não está presente, o valor de motion_vectors_over_pic_boundaries_flag deve ser inferido para ser igual a 1.

max_bytes_per_pic_denom indica um número de bytes não excedido pela soma dos tamanhos das unidades de camada de abstração de rede (NAL) de camada de codificação virtual (VCL) associadas com qualquer imagem codificada na sequência de vídeo codificado.

O número de bytes que representam uma imagem no fluxo de unidade de camada de abstração de rede é especificado para este propósito como o número total de bytes de dados de unidade de camada de abstração de rede de camada de codificação virtual (isto é, o total das variáveis NumBytesInNALunit para as unidades de camada de abstração de rede de camada de codificação virtual) para a imagem. O valor de max_bytes_per_pic_denom deve estar na faixa de 0 a 16, inclusivo.

Dependendo do max_bytes_per_pic_denom o seguinte se aplica:

- Se max_bytes_per_pic_denom for igual a 0, então nenhum limite é indicado.

- De outro modo (max_bytes_per_pic_denom não é igual a 0), nenhuma imagem codificada deve ser representada na sequência de vídeo codificado por mais que o seguinte número de bytes:

(PicSizeInMbs * RawMbBits) ÷ (8 * max_bytes_per_pic_denom)

Quando o elemento de sintaxe max_bytes_per_pic_denom não está presente, o valor de max_bytes_per_pic_denom deve ser inferido para ser igual a 2. A variável PicSizeInMbs é o número de macroblocos na imagem. A variável RawMbBits é derivada tal como na subcláusula 7.4.2.1 do Padrão MPEG-4 AVC.

max_bytes_per_mb_denom indica o número máximo de bits codificados dos dados macroblock_layer() para qualquer macrobloco em qualquer imagem da sequência de vídeo codificado. O valor de max_bytes_per_mb_denom deve estar na faixa de 0 a 16, inclusivo.

Dependendo do max_bytes_per_mb_denom o seguinte se aplica:

- Se max_bytes_per_mb_denom for igual a 0, então nenhum limite é especificado.

- De outro modo (max_bytes_per_mb_denom não é igual a 0), nenhum macroblock_layer() codificado deve ser representado no fluxo de bits por mais que o seguinte número de bits:

(128 + RawMbBits) ÷ max_bytes_per_mb_denom

Dependendo do entropy_coding_mode_flag, os bits dos dados de macroblock_layer() são contados como se segue:

- Se entropy_coding_mode_flag for igual a 0, então o número de bits dos dados de macroblock_layer() é dado pelo número de bits na estrutura de sintaxe macroblock_layer() para um macrobloco.

- De outro modo (entropy_coding_mode_flag é igual a 1), o número de bits dos dados de macroblock_layer() para um macrobloco é dado pelo número de vezes que read_bits(1) é chamado nas subcláusulas 9.3.3.2.2 e 9.3.3.2.3 do Padrão MPEG-4 AVC quando analisando sintaticamente o macroblock_layer() associado com o macrobloco.

Quando o max_bytes_per_mb_denom não está presente, o valor de max_bytes_per_mb_denom deve ser inferido para ser igual a 1.

log2_max_mv_length_horizontal e log2_max_mv_length_vertical indicam o valor absoluto máximo de um componente de vetor de movimento horizontal e vertical decodificado, respectivamente, em unidades de amostra de luminância de 1/4, para todas as imagens na sequência de vídeo codificado. Um valor de n expressa que nenhum valor de um componente de vetor de movimento deve exceder a faixa de -2^n a 2^n-1, inclusivo, em unidades de deslocamento de amostra de luminância de 1/4. O valor de log2_max_mv_length_horizontal deve estar na faixa de 0 a 16, inclusivo. O valor de log2_max_mv_length_vertical deve estar na faixa de 0 a 16, inclusivo. Quando log2_max_mv_length_horizontal não está presente, os valores de log2_max_mv_length_horizontal e log2_max_mv_length_vertical devem ser inferidos para serem iguais a 16. É para ser notado que o valor absoluto máximo de um componente de vetor de movimento vertical ou horizontal decodificado também é restringido por perfil e limites de nível tal como especificado no Anexo A do Padrão MPEG-4 AVC.

num_reorder_frames indica o número máximo de quadros, pares de campos complementares, ou campos não casados que precedem respectivamente qualquer quadro, par de campos complementares, ou campo não casado na sequência de vídeo codificado na ordem de decodificação e o segue na ordem de saída. O valor de num_reorder_frames deve estar na faixa de 0 a max_dec_frame_buffering, inclusivo. Quando o elemento de sintaxe num_reorder_frames não está presente, o valor de num_reorder_frames deve ser inferido como se segue:

- Se profile_idc for igual a 44, 100, 110, 122, ou 244 e constraint_set3_flag for igual a 1, então o valor de num_reorder_frames deve ser inferido para ser igual a 0.

- De outro modo (profile_idc não é igual a 44, 100, 110, 122, ou 244 ou constraint_set3_flag é igual a 0), o valor de num_reorder_frames deve ser inferido para ser igual a max_dec_frame_bufferingMaxDpbSize.

max_dec_frame_buffering especifica o tamanho exigido do armazenamento temporário de imagem decodificada (DPB) de decodificador hipotético de referência em unidades

de armazenamentos temporários de quadro. A sequência de vídeo codificado não deve exigir um armazenamento temporário de imagem decodificada com tamanho de mais que Max(1, max_dec_frame_buffering) armazenamentos temporários de quadro para capacitar a saída de imagens decodificadas nos tempos de saída especificados por dpb_output_delay das mensagens de Informação Suplementar de Enriquecimento (SEI) de sincronismo de imagem. O valor de max_dec_frame_buffering deve estar na faixa de num_ref_frames a MaxDpbSize (tal como especificado na subcláusula A.3.1 ou A.3.2 do Padrão MPEG-4 AVC), inclusivo. Quando o elemento de sintaxe max_dec_frame_buffering não está presente, o valor de max_dec_frame_buffering deve ser inferido como se segue:

- Se profile_idc for igual a 44 ou 244 e constraint_set3_flag for igual a 1, então o valor de max_dec_frame_buffering deve ser inferido para ser igual a 0.

- De outro modo (profile_idc não é igual a 44 ou 244 ou constraint_set3_flag é igual a 0), o valor de max_dec_frame_buffering deve ser inferido para ser igual a MaxDpbSize.

Em codificação de vídeo de múltiplas visualizações, os parâmetros de restrição de fluxo de bits customizam a operação de decodificação de um subfluxo com base em limites mais apertados. Portanto, aos parâmetros de restrição de fluxo de bits deve ser permitido serem especificados para cada subfluxo extraível de um fluxo de bits codificado de vídeo de múltiplas visualizações. De acordo com uma modalidade, propomos especificar informação de restrição de fluxo de bits para cada visualização, para cada nível temporal em uma visualização e/ou para cada ponto de operação.

Especificação de parâmetros de restrição de fluxo de bits para cada visualização.

Parâmetros de restrição de fluxo de bits podem ser especificados para cada visualização. Propomos a sintaxe de mvc_vui_parameters_extension, a qual é uma parte da subset_sequence_parameter_set. A TABELA 2 ilustra a sintaxe de mvc_vui_parameters_extension.

mvc_vui_parameters_extension() executa laços sobre todas as visualizações que estão associadas a esta subset_sequence_parameter_set. A view_id de cada visualização e os parâmetros de restrição de fluxo de bits de cada visualização são especificados dentro do laço.

TABELA 2

mvc_vui_parameters_extension(){	C	Descritor
num_views_minus1	0	ue(v)
for(i=0; i<=num_views_minus1; i++){		
view_id[i]	0	u(3)
bitstream_restriction_flag[i]	0	u(1)
if(bitstream_restriction_flag[i]){		
motion_vectors_over_pic_boundaries_flag[i]	0	u(1)

max_bytes_per_pic_denom[i]	0	ue(v)
max_bits_per_mb_denom[i]	0	ue(v)
log2_max_mv_length_horizontal[i]	0	ue(v)
log2_max_mv_length_vertical[i]	0	ue(v)
num_reorder_frames[i]	0	ue(v)
max_dec_frame_buffering[i]	0	ue(v)
}		
}		
}		

As semânticas dos elementos de sintaxe de restrição de fluxo de bits são como se segue:

bitstream_restriction_flag[i] especifica o valor de bitstream_restriction_flag da visualização tendo view_id[i] igual a view_id.

motion_vectors_over_pic_boundaries_flag[i] especifica o valor de motion_vectors_over_pic_boundaries_flag da visualização tendo view_id[i] igual a view_id. Quando o elemento de sintaxe motion_vectors_over_pic_boundaries_flag[i] não está presente, o valor de motion_vectors_over_pic_boundaries_flag para a visualização tendo view_id[i] igual a view_id deve ser inferido para ser igual a 1.

max_bytes_per_pic_denom[i] especifica o valor de max_bytes_per_pic_denom da visualização tendo view_id[i] igual a view_id. Quando o elemento de sintaxe max_bytes_per_pic_denom[i] não está presente, o valor de max_bytes_per_pic_denom da visualização tendo view_id[i] igual a view_id deve ser inferido para ser igual a 2.

max_bytes_per_mb_denom[i] especifica o valor de max_bytes_per_mb_denom da visualização tendo view_id[i] igual a view_id. Quando o max_bytes_per_mb_denom[i] não está presente, o valor de max_bytes_per_mb_denom da visualização tendo view_id[i] igual a view_id deve ser inferido para ser igual a 1.

log2_max_mv_length_horizontal[i] e log2_max_mv_length_vertical[i] especificam respectivamente os valores de log2_max_mv_length_horizontal e log2_max_mv_length_vertical da visualização tendo view_id[i] igual a view_id. Quando log2_max_mv_length_horizontal[i] não está presente, os valores de log2_max_mv_length_horizontal e log2_max_mv_length_vertical da visualização tendo view_id[i] igual a view_id devem ser inferidos para serem iguais a 16.

num_reorder_frames[i] especifica o valor de num_reorder_frames da visualização tendo view_id[i] igual a view_id. O valor de num_reorder_frames[i] deve estar na faixa de 0 a max_dec_frame_buffering, inclusivo. Quando o elemento de sintaxe num_reorder_frames[i] não está presente, o valor de num_reorder_frames da visualização

tendo view_id[i] igual a view_id deve ser inferido para ser igual a max_dec_frame_buffering.

max_dec_frame_buffering[i] especifica o valor de max_dec_frame_buffering da visualização tendo view_id[i] igual a view_id. O valor de max_dec_frame_buffering[i] deve estar na faixa de num_ref_frames[i] a MaxDpbSize (tal como especificado na subcláusula A.3.1 ou A.3.2 no Padrão MPEG-4 AVC), inclusivo. Quando o elemento de sintaxe max_dec_frame_buffering[i] não está presente, o valor de max_dec_frame_buffering da visualização tendo view_id[i] igual a view_id deve ser inferido para ser igual a MaxDpbSize.

Voltando à figura 3, um método exemplar para codificar parâmetros de restrição de fluxo de bits para cada visualização, usando um elemento de sintaxe mvc_vui_parameters_extension(), está indicado de uma maneira geral pelo número de referência 300.

O método 300 inclui um bloco de início 305 que passa o controle para um bloco de função 310. O bloco de função 310 estabelece uma variável M igual a um número de visualizações menos um, e passa o controle para um bloco de função 315. O bloco de função 315 grava a variável M para um fluxo de bits, e passa o controle para um bloco de função 320. O bloco de função 320 estabelece uma variável i igual a zero, e passa o controle para um bloco de função 325. O bloco de função 325 grava um elemento de sintaxe view_id[i], e passa o controle para um bloco de função 330. O bloco de função 330 grava um elemento de sintaxe bitstream_restriction_flag[i], e passa o controle para um bloco de decisão 335. O bloco de decisão 335 determina se o elemento de sintaxe bitstream_restriction_flag[i] é ou não igual a zero. Neste caso, então o controle é passado para um bloco de decisão 345. De outro modo, o controle é passado para um bloco de função 340.

O bloco de função 340 grava os parâmetros de restrição de fluxo de bits da visualização i, e passa o controle para o bloco de decisão 345. O bloco de decisão 345 determina se a variável i é ou não igual à variável M. Neste caso, o controle é então passado para um bloco final 399. De outro modo, o controle é passado para um bloco de função 350.

O bloco de função 350 estabelece a variável i igual a i mais um, e retorna o controle para o bloco de função 325.

Voltando à figura 4, um método exemplar para decodificar parâmetros de restrição de fluxo de bits para cada visualização, usando um elemento de sintaxe mvc_vui_parameters_extension(), está indicado de uma maneira geral pelo número de referência 400.

O método 400 inclui um bloco de início 405 que passa o controle para um bloco de função 407. O bloco de função 407 lê uma variável M de um fluxo de bits, e passa o controle para um bloco de função 410. O bloco de função 410 estabelece o número de visualizações igual à variável M mais um, e passa o controle para um bloco de função 420. O bloco de

função 420 estabelece uma variável i igual a zero, e passa o controle para um bloco de função 425. O bloco de função 425 lê um elemento de sintaxe view_id[i], e passa o controle para um bloco de função 430. O bloco de função 430 lê um elemento de sintaxe bitstream_restriction_flag[i], e passa o controle para um bloco de decisão 435. O bloco de decisão 435 determina se o elemento de sintaxe bitstream_restriction_flag[i] é ou não igual a zero. Neste caso, o controle é então passado para um bloco de decisão 445. De outro modo, o controle é passado para um bloco de função 440.

O bloco de função 440 lê os parâmetros de restrição de fluxo de bits da visualização i, e passa o controle para o bloco de decisão 445. O bloco de decisão 445 determina se a variável i é ou não igual à variável M. Neste caso, o controle é então passado para um bloco final 499. De outro modo, o controle é passado para um bloco de função 450.

O bloco de função 450 estabelece a variável i igual a i mais um, e retorna o controle para o bloco de função 425.

Especificação de parâmetros de restrição de fluxo de bits para cada nível temporal de cada visualização.

Parâmetros de restrição de fluxo de bits podem ser especificados para cada nível temporal de cada visualização. Propomos a sintaxe de mvc_vui_parameters_extension, a qual é uma parte da subset_sequence_parameter_set. A TABELA 3 ilustra a sintaxe de mvc_vui_parameters_extension.

TABELA 3

mvc_vui_parameters_extension(){	C	Descritor
num_views_minus1	0	ue(v)
for(i=0; i<=num_views_minus1; i++){		
view_id[i]	0	u(3)
num_temporal_layers_in_view_minus1[i]	0	ue(v)
for(j=0; j<= num_temporal_level_in_view_minus1; j++){		
temporal_id[i][j]		
bitstream_restriction_flag[i][j]	0	u(1)
if(bitstream_restriction_flag[i][j]){		
motion_vectors_over_pic_boundaries_flag[i][j]	0	u(1)
max_bytes_per_pic_denom[i][j]	0	ue(v)
max_bits_per_mb_denom[i][j]	0	ue(v)
log2_max_mv_length_horizontal[i][j]	0	ue(v)
log2_max_mv_length_vertical[i][j]	0	ue(v)
num_reorder_frames[i][j]	0	ue(v)
max_dec_frame_buffering[i][j]	0	ue(v)
}		

}		
}		
}		

As semânticas dos elementos de sintaxe de restrição de fluxo de bits são como se segue:

bitstream_restriction_flag[i][j] especifica o valor de fluxo de bits restriction_flag do nível temporal tendo temporal_id[i][j] igual a temporal_id na visualização tendo view_id[i] igual a view_id.

motion_vectors_over_pic_boundaries_flag[i][j] especifica o valor de motion_vectors_over_pic_boundaries_flag do nível temporal tendo temporal_id[i][j] igual a temporal_id na visualização tendo view_id[i] igual a view_id. Quando o elemento de sintaxe motion_vectors_over_pic_boundaries_flag[i] não está presente, o valor de motion_vectors_over_pic_boundaries_flag do nível temporal tendo temporal_id[i][j] igual a temporal_id na visualização tendo view_id[i] igual a view_id deve ser inferido para ser igual a 1.

max_bytes_per_pic_denom[i][j] especifica o valor de max_bytes_per_pic_denom do nível temporal tendo temporal_id[i][j] igual a temporal_id na visualização tendo view_id[i] igual a view_id. Quando o elemento de sintaxe max_bytes_per_pic_denom[i] não está presente, o valor de max_bytes_per_pic_denom do nível temporal tendo temporal_id[i][j] igual a temporal_id na visualização tendo view_id[i] igual a view_id deve ser inferido para ser igual a 2.

max_bytes_per_mb_denom[i][j] especifica o valor de max_bytes_per_mb_denom do nível temporal tendo temporal_id[i][j] igual a temporal_id na visualização tendo view_id[i] igual a view_id. Quando o max_bytes_per_mb_denom[i] não está presente, o valor de max_bytes_per_mb_denom do nível temporal tendo temporal_id[i][j] igual a temporal_id na visualização tendo view_id[i] igual a view_id deve ser inferido para ser igual a 1.

log2_max_mv_length_horizontal[i][j] e log2_max_mv_length_vertical[i][j] especificam respectivamente os valores de log2_max_mv_length_horizontal e log2_max_mv_length_vertical do nível temporal tendo temporal_id[i][j] igual a temporal_id na visualização tendo view_id[i] igual a view_id. Quando log2_max_mv_length_horizontal[i] não está presente, os valores de log2_max_mv_length_horizontal e log2_max_mv_length_vertical do nível temporal tendo temporal_id[i][j] igual a temporal_id na visualização tendo view_id[i] igual a view_id devem ser inferidos para serem iguais a 16.

num_reorder_frames[i][j] especifica o valor de num_reorder_frames do nível temporal tendo temporal_id[i][j] igual a temporal_id na visualização tendo view_id[i] igual a view_id. O valor de num_reorder_frames[i] deve estar na faixa de 0 a

max_dec_frame_buffering, inclusivo. Quando o elemento de sintaxe num_reorder_frames[i] não está presente, o valor de num_reorder_frames do nível temporal tendo temporal_id[i][j] igual a temporal_id na visualização tendo view_id[i] igual a view_id deve ser inferido para ser igual a max_dec_frame_buffering.

max_dec_frame_buffering[i][j] especifica o valor de max_dec_frame_buffering do nível temporal tendo temporal_id[i][j] igual a temporal_id na visualização tendo view_id[i] igual a view_id. O valor de max_dec_frame_buffering[i] deve estar na faixa de num_ref_frames[i] a MaxDpbSize (tal como especificado na subcláusula A.3.1 ou A.3.2 no Padrão MPEG-4 AVC), inclusivo. Quando o elemento de sintaxe max_dec_frame_buffering[i] não está presente, o valor de max_dec_frame_buffering do nível temporal tendo temporal_id[i][j] igual a temporal_id na visualização tendo view_id[i] igual a view_id deve ser inferido para ser igual a MaxDpbSize.

Em mvc_vui_parameters_extension(), dois laços são executados. O laço externo executa ciclo sobre todas as visualizações associadas ao subset_sequence_parameter_set. O view_id para o número de níveis temporais de cada visualização é especificado no laço externo. O laço interno executa ciclo sobre todos os níveis temporais de uma visualização. A informação de restrição de fluxo de bits é especificada no laço interno.

Voltando à figura 5, um método exemplar para codificar parâmetros de restrição de fluxo de bits para cada nível temporal em cada visualização, usando um elemento de sintaxe mvc_vui_parameters_extension(), está indicado de uma maneira geral pelo número de referência 500.

O método 500 inclui um bloco de início 505 que passa o controle para um bloco de função 510. O bloco de função 510 estabelece uma variável M igual a um número de visualizações menos um, e passa o controle para um bloco de função 515. O bloco de função 515 grava a variável M para um fluxo de bits, e passa o controle para um bloco de função 520. O bloco de função 520 estabelece uma variável i igual a zero, e passa o controle para um bloco de função 525. O bloco de função 525 grava um elemento de sintaxe view_id[i], e passa o controle para um bloco de função 530. O bloco de função 530 estabelece uma variável N igual a um número de níveis temporais na visualização i menos 1, e passa o controle para um bloco de função 535. O bloco de função 535 grava a variável N para o fluxo de bits, e passa o controle para um bloco de função 540. O bloco de função 540 estabelece uma variável j igual a zero, e passa o controle para um bloco de função 545. O bloco de função 545 grava um elemento de sintaxe temporal_id[i][j], e passa o controle para um bloco de função 550. O bloco de função 550 grava um elemento de sintaxe bitstream_restriction_flag[i][j], e passa o controle para um bloco de decisão 555. O bloco de decisão 555 determina se o elemento de sintaxe bitstream_restriction_flag[i][j] é ou não igual a zero. Neste caso, o controle é então passado para um bloco de decisão 565. De outro modo, o controle é passado

para um bloco de função 560.

O bloco de função 560 grava os parâmetros de restrição de fluxo de bits de nível temporal j na visualização i, e passa o controle para o bloco de decisão 565. O bloco de decisão 565 determina se a variável j é ou não igual à variável N. Neste caso, o controle é então passado para um bloco de decisão 570. De outro modo, o controle é passado para um bloco de função 575.

O bloco de decisão 570 determina se a variável i é ou não igual à variável M. Neste caso, o controle é então passado para um bloco final 599. De outro modo, o controle é passado para um bloco de função 580.

O bloco de função 580 estabelece a variável i igual a i mais um, e retorna o controle para o bloco de função 525.

O bloco de função 575 estabelece a variável j igual a j mais um, e retorna o controle para o bloco de função 545.

Voltando à figura 6, um método exemplar para decodificar parâmetros de restrição de fluxo de bits para cada nível temporal em cada visualização, usando um elemento de sintaxe mvc_vui_parameters_extension(), está indicado de uma maneira geral pelo número de referência 600.

O método 600 inclui um bloco de início 605 que passa o controle para um bloco de função 607. O bloco de função 607 lê uma variável M de um fluxo de bits, e passa o controle para um bloco de função 610. O bloco de função 610 estabelece um número de visualizações igual a M mais um, e passa o controle para um bloco de função 620. O bloco de função 620 estabelece uma variável i igual a zero, e passa o controle para um bloco de função 625. O bloco de função 625 lê um elemento de sintaxe view_id[i], e passa o controle para um bloco de função 627. O bloco de função 627 lê uma variável N do fluxo de bits, e passa o controle para um bloco de função 630. O bloco de função 630 estabelece um número de níveis temporais na visualização i igual a N mais 1, e passa o controle para um bloco de função 640. O bloco de função 640 estabelece uma variável j igual a zero, e passa o controle para um bloco de função 645. O bloco de função 645 lê um elemento de sintaxe temporal_id[i][j], e passa o controle para um bloco de função 650. O bloco de função 650 lê um elemento de sintaxe bitstream_restriction_flag[i][j], e passa o controle para um bloco de decisão 655. O bloco de decisão 655 determina se o elemento de sintaxe bitstream_restriction_flag[i][j] é ou não igual a zero. Neste caso, o controle é então passado para um bloco de decisão 665. De outro modo, o controle é passado para um bloco de função 660.

O bloco de função 660 lê os parâmetros de restrição de fluxo de bits de nível temporal j na visualização i, e passa o controle para o bloco de decisão 665. O bloco de decisão 665 determina se a variável j é ou não igual à variável N. Neste caso, o controle é então

passado para um bloco de decisão 670. De outro modo, o controle é passado para um bloco de função 675.

O bloco de decisão 670 determina se a variável i é ou não igual à variável M. Neste caso, o controle é então passado para um bloco final 699. De outro modo, o controle é passado para um bloco de função 680.

O bloco de função 680 estabelece a variável i igual a i mais um, e retorna o controle para o bloco de função 625.

O bloco de função 675 estabelece a variável j igual a j mais um, e retorna o controle para o bloco de função 645.

Especificação de informação de restrição de fluxo de bits para cada ponto de operação.

Parâmetros de restrição de fluxo de bits podem ser especificados para cada ponto de operação. Propomos transportar os parâmetros de restrição de fluxo de bits de cada ponto de operação na mensagem SEI de informação de escalabilidade de visualização. A sintaxe de mensagem SEI de informação de escalabilidade de visualização pode ser modificada tal como na TABELA 4. A sintaxe para informação de restrição de fluxo de bits é inserida em um laço que executa ciclo sobre todos os pontos de operação.

TABELA 4

view_scalability_info(payloadSize){	C	Descritor
num_operation_points_minus1	5	ue(v)
for(i = 0; i <= num_operation_points_minus1; i++){		
operation_point_id[i]	5	ue(v)
priority_id[i]	5	u(5)
temporal_id[i]	5	u(3)
num_active_views_minus1[i]	5	ue(v)
for(j = 0; j <= num_active_views_minus1[i]; j++)		
view_id[i][j]	5	ue(v)
profile_level_info_present_flag[i]	5	u(1)
bitrate_info_present_flag[i]	5	u(1)
frm_rate_info_present_flag[i]	5	u(1)
op_dependency_info_present_flag[i]	5	u(1)
init_parameter_sets_info_present_flag[i]	5	u(1)
bitstream_restriction_flag[i]		
if(profile_level_info_present_flag[i]){		
op_profile_idc[i]	5	u(8)
op_constraint_set0_flag[i]	5	u(1)
op_constraint_set1_flag[i]	5	u(1)

op_constraint_set2_flag[i]	5	u(1)
op_constraint_set3_flag[i]	5	u(1)
reserved_zero_4bits /* equal to 0 */	5	u(4)
op_level_idc[i]	5	u(8)
} else		
profile_level_info_src_op_id_delta[i]		ue(v)
if(bitrate_info_present_flag[i]){		
avg_bitrate[i]	5	u(16)
max_bitrate[i]	5	u(16)
max_bitrate_calc_window[i]	5	u(16)
}		
if(frm_rate_info_present_flag[i]){		
constant_frm_rate_idc[i]	5	u(2)
avg_frm_rate[i]	5	u(16)
} else		
frm_rate_info_src_op_id_delta[i]	5	ue(v)
if(op_dependency_info_present_flag[i]){		
num_directly_dependent_ops[i]	5	ue(v)
for(j = 0; j < num_directly_dependent_ops[i]; j++){		
directly_dependent_op_id_delta_minus1[i][j]	5	ue(v)
} else		
op_dependency_info_src_op_id_delta[i]	5	ue(v)
if(init_parameter_sets_info_present_flag[i]){		
num_init_seq_parameter_set_minus1[i]	5	ue(v)
for(j = 0; j <= num_init_seq_parameter_set_minus1[i]; j++)		
init_seq_parameter_set_id_delta[i][j]	5	ue(v)
num_init_pic_parameter_set_minus1[i]	5	ue(v)
for(j = 0; j <= num_init_pic_parameter_set_minus1[i]; j++)		
init_pic_parameter_set_id_delta[i][j]	5	ue(v)
} else		
init_parameter_sets_info_src_op_id_delta[i]	5	ue(v)
if(bitstream_restriction_flag[i]){		
motion_vectors_over_pic_boundaries_flag[i]	0	u(1)
max_bytes_per_pic_denom[i]	0	ue(v)
max_bits_per_mb_denom[i]	0	ue(v)
log2_max_mv_length_horizontal[i]	0	ue(v)

log2_max_mv_length_vertical[i]	0	ue(v)
num_reorder_frames[i]	0	ue(v)
max_dec_frame_buffering[i]	0	ue(v)
}		
}		
}		

As semânticas dos elementos de sintaxe de restrição de fluxo de bits são como se segue:

bitstream_restriction_flag[i] especifica o valor de bitstream_restriction_flag do ponto de operação tendo operation_point_id[i] igual a operation_point_id.

motion_vectors_over_pic_boundaries_flag[i] especifica o valor de motion_vectors_over_pic_boundaries_flag do ponto de operação tendo operation_point_id[i] igual a operation_point_id. Quando o elemento de sintaxe motion_vectors_over_pic_boundaries_flag[i] não está presente, o valor de motion_vectors_over_pic_boundaries_flag do ponto de operação tendo operation_point_id[i] igual a operation_point_id deve ser inferido para ser igual a 1.

max_bytes_per_pic_denom[i] especifica o valor de max_bytes_per_pic_denom do ponto de operação tendo operation_point_id[i] igual a operation_point_id. Quando o elemento de sintaxe max_bytes_per_pic_denom[i] não está presente, o valor de max_bytes_per_pic_denom do ponto de operação tendo operation_point_id[i] igual a operation_point_id deve ser inferido para ser igual a 2.

max_bytes_per_mb_denom[i] especifica o valor de max_bytes_per_mb_denom do ponto de operação tendo operation_point_id[i] igual a operation_point_id. Quando o max_bytes_per_mb_denom[i] não está presente, o valor de max_bytes_per_mb_denom do ponto de operação tendo operation_point_id[i] igual a operation_point_id deve ser inferido para ser igual a 1.

log2_max_mv_length_horizontal[i] e log2_max_mv_length_vertical[i] especificam respectivamente o valor de log2_max_mv_length_horizontal e o valor de log2_max_mv_length_vertical do ponto de operação tendo operation_point_id[i] igual a operation_point_id. Quando log2_max_mv_length_horizontal[i] não está presente, os valores de log2_max_mv_length_horizontal e log2_max_mv_length_vertical do ponto de operação tendo operation_point_id[i] igual a operation_point_id devem ser inferidos para serem iguais a 16.

num_reorder_frames[i] especifica o valor de num_reorder_frames do ponto de operação tendo operation_point_id[i] igual a operation_point_id. O valor de num_reorder_frames[i] deve estar na faixa de 0 a max_dec_frame_buffering, inclusivo.

Quando o elemento de sintaxe num_reorder_frames[i] não está presente, o valor de num_reorder_frames do ponto de operação tendo operation_point_id[i] igual a operation_point_id deve ser inferido para ser igual a max_dec_frame_buffering.

max_dec_frame_buffering[i] especifica o valor de max_dec_frame_buffering do ponto de operação tendo operation_point_id[i] igual a operation_point_id. O valor de max_dec_frame_buffering[i] deve estar na faixa de num_ref_frames[i] a MaxDpbSize (tal como especificado na subcláusula A.3.1 ou A.3.2 no Padrão MPEG-4 AVC), inclusivo. Quando o elemento de sintaxe max_dec_frame_buffering[i] não está presente, o valor de max_dec_frame_buffering do ponto de operação tendo operation_point_id[i] igual a operation_point_id deve ser inferido para ser igual a MaxDpbSize.

Voltando à figura 7, um método exemplar para codificar parâmetros de restrição de fluxo de bits para cada ponto de operação, usando um elemento de sintaxe view_scalability_parameters_extension(), está indicado de uma maneira geral pelo número de referência 700.

O método 700 inclui um bloco de início 705 que passa o controle para um bloco de função 710. O bloco de função 710 estabelece uma variável M igual a um número de pontos de operação menos um, e passa o controle para um bloco de função 715. O bloco de função 715 grava a variável M para um fluxo de bits, e passa o controle para um bloco de função 720. O bloco de função 720 estabelece uma variável i igual a zero, e passa o controle para um bloco de função 725. O bloco de função 725 grava um elemento de sintaxe operation_point_id[i], e passa o controle para um bloco de função 730. O bloco de função 730 grava um elemento de sintaxe bitstream_restriction_flag[i], e passa o controle para um bloco de decisão 735. O bloco de decisão 735 determina se o elemento de sintaxe bitstream_restriction_flag[i] é ou não igual a zero. Neste caso, o controle é então passado para um bloco de decisão 745. De outro modo, o controle é passado para um bloco de função 740.

O bloco de função 740 grava os parâmetros de restrição de fluxo de bits do ponto de operação i, e passa o controle para o bloco de decisão 745. O bloco de decisão 745 determina se a variável i é ou não igual à variável M. Neste caso, o controle é então passado para um bloco final 799. De outro modo, o controle é passado para um bloco de função 750.

O bloco de função 750 estabelece a variável i igual a i mais um, e retorna o controle para o bloco de função 725.

Voltando à figura 8, um método exemplar para decodificar parâmetros de restrição de fluxo de bits para cada ponto de operação, usando um elemento de sintaxe view_scalability_parameters_extension(), está indicado de uma maneira geral pelo número de referência 800.

O método 800 inclui um bloco de início 805 que passa o controle para um bloco de função 807. O bloco de função 807 lê uma variável M de um fluxo de bits, e passa o controle

para um bloco de função 810. O bloco de função 810 estabelece um número de pontos de operação igual a M mais um, e passa o controle para um bloco de função 820. O bloco de função 820 estabelece uma variável i igual a zero, e passa o controle para um bloco de função 825. O bloco de função 825 lê um elemento de sintaxe operation_point_id[i], e passa o controle para um bloco de função 830. O bloco de função 830 lê um elemento de sintaxe bitstream_restriction_flag[i], e passa o controle para um bloco de decisão 835. O bloco de decisão 835 determina se o elemento de sintaxe bitstream_restriction_flag[i] é ou não igual a zero. Neste caso, o controle é então passado para um bloco de decisão 845. De outro modo, o controle é passado para um bloco de função 840.

O bloco de função 840 lê os parâmetros de restrição de fluxo de bits de ponto de operação i, e passa o controle para o bloco de decisão 845. O bloco de decisão 845 determina se a variável i é ou não igual à variável M. Neste caso, o controle é então passado para um bloco final 899. De outro modo, o controle é passado para um bloco de função 850.

O bloco de função 850 estabelece a variável i igual a i mais um, e retorna o controle para o bloco de função 825.

Uma descrição será dada agora de algumas das muitas vantagens/recursos concomitantes da presente invenção, alguns dos quais foram mencionados anteriormente. Por exemplo, uma vantagem/recurso é um aparelho que inclui um codificador para codificar conteúdo de vídeo de múltiplas visualizações ao especificar Informação de Usabilidade de Vídeo para pelo menos um de visualizações individuais, níveis temporais individuais em uma visualização e pontos de operação individuais.

Uma outra vantagem/recurso é o aparelho tendo o codificador tal como descrito anteriormente, em que os parâmetros são especificados em pelo menos um elemento de sintaxe de alto nível.

Além disso, uma outra vantagem/recurso é o aparelho tendo o codificador tal como descrito anteriormente, em que o pelo menos elemento de sintaxe de alto nível inclui pelo menos um de um elemento de sintaxe mvc_vui_parameters_extension(), uma mensagem de sintaxe de informação suplementar de enriquecimento mvc_scalability_info, pelo menos uma parte de um conjunto de parâmetros de sequência, um conjunto de parâmetros de imagem e informação suplementar de enriquecimento.

Adicionalmente, uma outra vantagem/recurso é o aparelho tendo o codificador tal como descrito anteriormente, em que pelo menos uma parte da Informação de Usabilidade de Vídeo compreende parâmetros de restrição de fluxo de bits.

Estes e outros recursos e vantagens dos presentes princípios podem ser prontamente apurados por uma pessoa de conhecimento comum na técnica pertinente com base nos preceitos neste documento. É para ser entendido que os preceitos dos presentes princípios podem ser implementados em várias formas de hardware, software, firmware, proces-

sadores de uso especial ou combinações dos mesmos.

Mais preferivelmente, os preceitos dos presentes princípios são implementados como uma combinação de hardware e software. Além disso, o software pode ser implementado como um programa de aplicação incorporado de modo tangível em uma unidade de armazenamento de programa. O programa de aplicação pode ser carregado e executado por uma máquina compreendendo qualquer arquitetura adequada. Preferivelmente, a máquina é implementada em uma plataforma de computador tendo hardware tal como uma ou mais unidades centrais de processamento ("CPU"), uma memória de acesso aleatório ("RAM") e interfaces de entrada/saída ("I/O"). A plataforma de computador também pode incluir um sistema de operação e código de microinstrução. Os vários processos e funções descritos neste documento podem ser parte do código de microinstrução ou parte do programa de aplicação, ou de qualquer combinação dos mesmos, os quais podem ser executados por uma CPU. Além do mais, várias outras unidades periféricas podem ser conectadas à plataforma de computador, tais como uma unidade de armazenamento de dados adicional e uma unidade de impressão.

É para ser entendido adicionalmente que, por causa de alguns dos componentes e métodos de sistema constituinte representados nos desenhos anexos serem preferivelmente implementados em software, as conexões reais entre os componentes de sistema ou os blocos de função de processo podem diferir dependendo da maneira na qual os presentes princípios são programados. Dados os preceitos neste documento, uma pessoa de conhecimento comum na técnica pertinente será capaz de considerar estas e implementações ou configurações similares dos presentes princípios.

Embora as modalidades ilustrativas tenham sido descritas neste documento com referência aos desenhos anexos, é para ser entendido que os presentes princípios não estão limitados a essas modalidades definidas, e que várias mudanças e modificações podem ser efetuadas nas mesmas por uma pessoa de conhecimento comum na técnica pertinente sem divergir do escopo ou espírito dos presentes princípios. Todas as tais mudanças e modificações são pretendidas para estarem incluídas no escopo dos presentes princípios tal como exposto nas reivindicações anexas.

REIVINDICAÇÃO

1. Método, **CARACTERIZADO** pelo fato de que compreende:

decodificar conteúdo de vídeo de múltiplas visualizações ao especificar informação de usabilidade de vídeo para níveis temporais individuais em uma visualização (600).

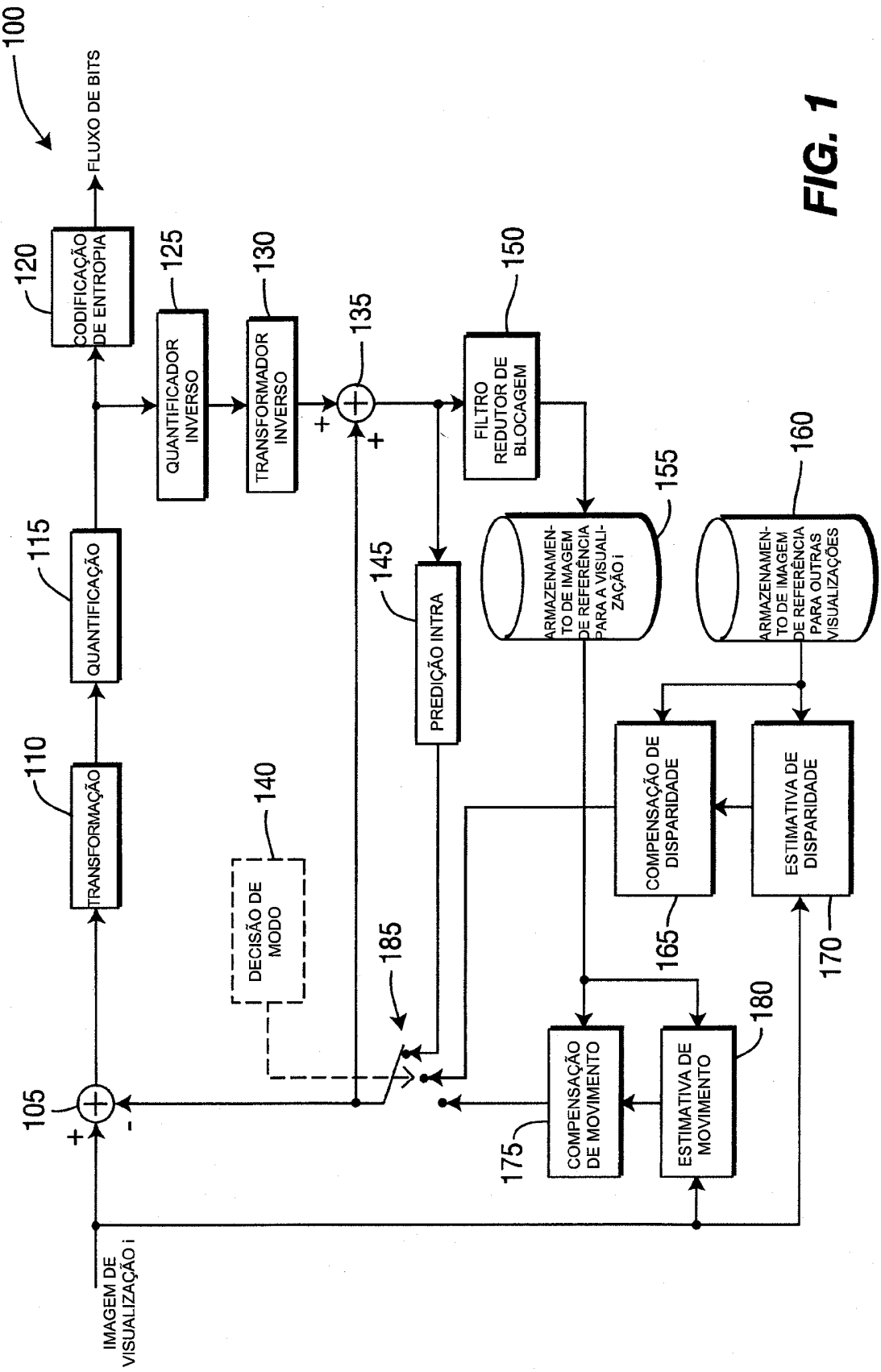

100
FLUXO DE BITS
120
CODIFICAÇÃO DE ENTROPIA
125
QUANTIFICADOR INVERSO
130
TRANSFORMADOR INVERSO
135
150
FILTRO REDUTOR DE BLOCAGEM
155
ARMAZENAMENTO DE IMAGEM DE REFERÊNCIA PARA A VISUALIZAÇÃO i
160
ARMAZENAMENTO DE IMAGEM DE REFERÊNCIA PARA OUTRAS VISUALIZAÇÕES
115
QUANTIFICAÇÃO
145
PREDIÇÃO INTRA
110
TRANSFORMAÇÃO
140
DECISÃO DE MODO
185
COMPENSAÇÃO DE DISPARIDADE
ESTIMATIVA DE DISPARIDADE
165
170
105
+
-
COMPENSAÇÃO DE MOVIMENTO
ESTIMATIVA DE MOVIMENTO
175
180
IMAGEM DE VISUALIZAÇÃO i

FIG. 1

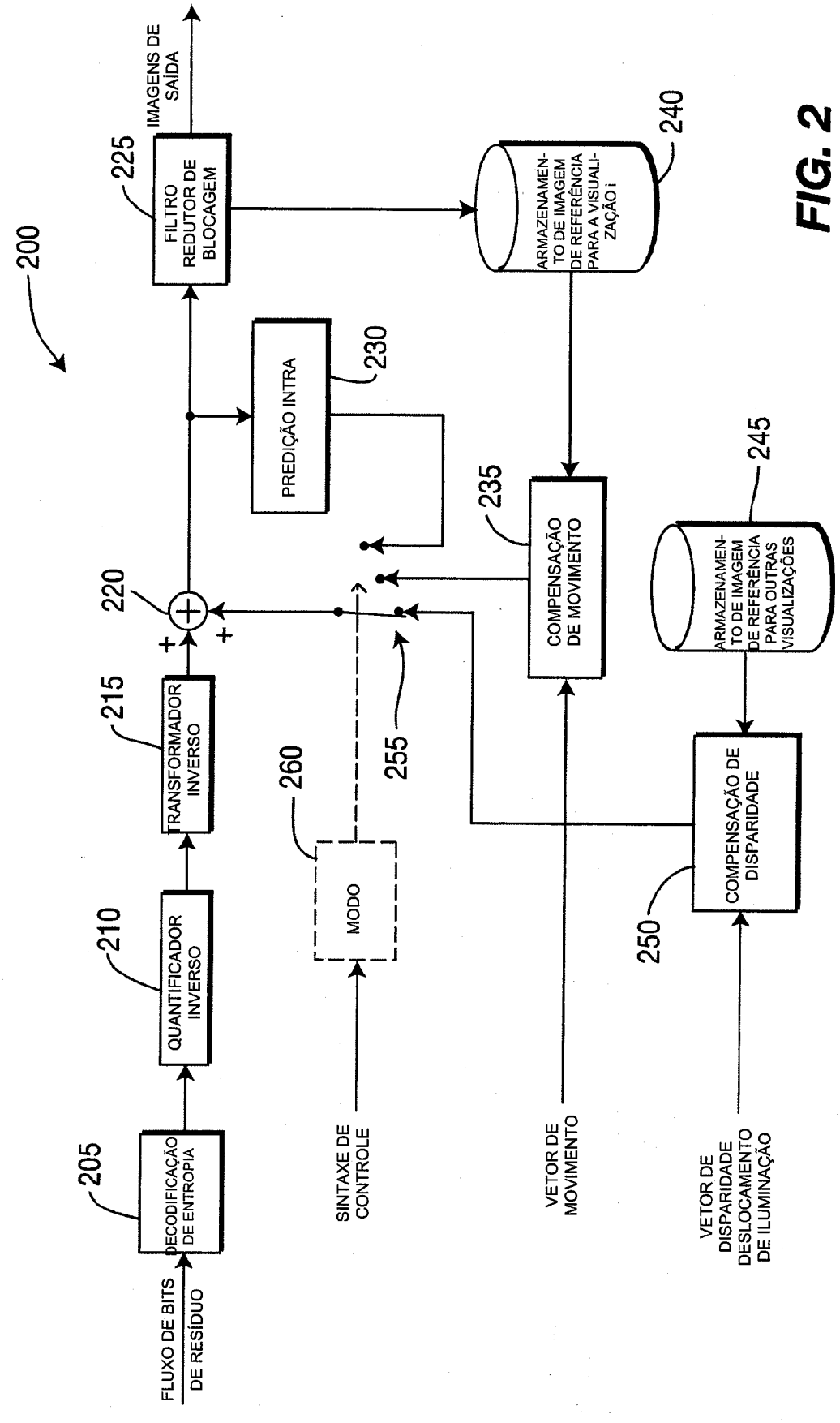

200
205
DECODIFICAÇÃO DE ENTROPIA
FLUXO DE BITS DE RESÍDUO
210
QUANTIFICADOR INVERSO
215
TRANSFORMADOR INVERSO
220
+
+
225
FILTRO REDUTOR DE BLOCAGEM
IMAGENS DE SAÍDA
230
PREDIÇÃO INTRA
235
COMPENSAÇÃO DE MOVIMENTO
240
ARMAZENAMENTO DE IMAGEM DE REFERÊNCIA PARA A VISUALIZAÇÃO i
245
ARMAZENAMENTO DE IMAGEM DE REFERÊNCIA PARA OUTRAS VISUALIZAÇÕES
250
COMPENSAÇÃO DE DISPARIDADE
255
260
MODO
SINTAXE DE CONTROLE
VETOR DE MOVIMENTO
VETOR DE DISPARIDADE DESLOCAMENTO DE ILUMINAÇÃO

FIG. 2

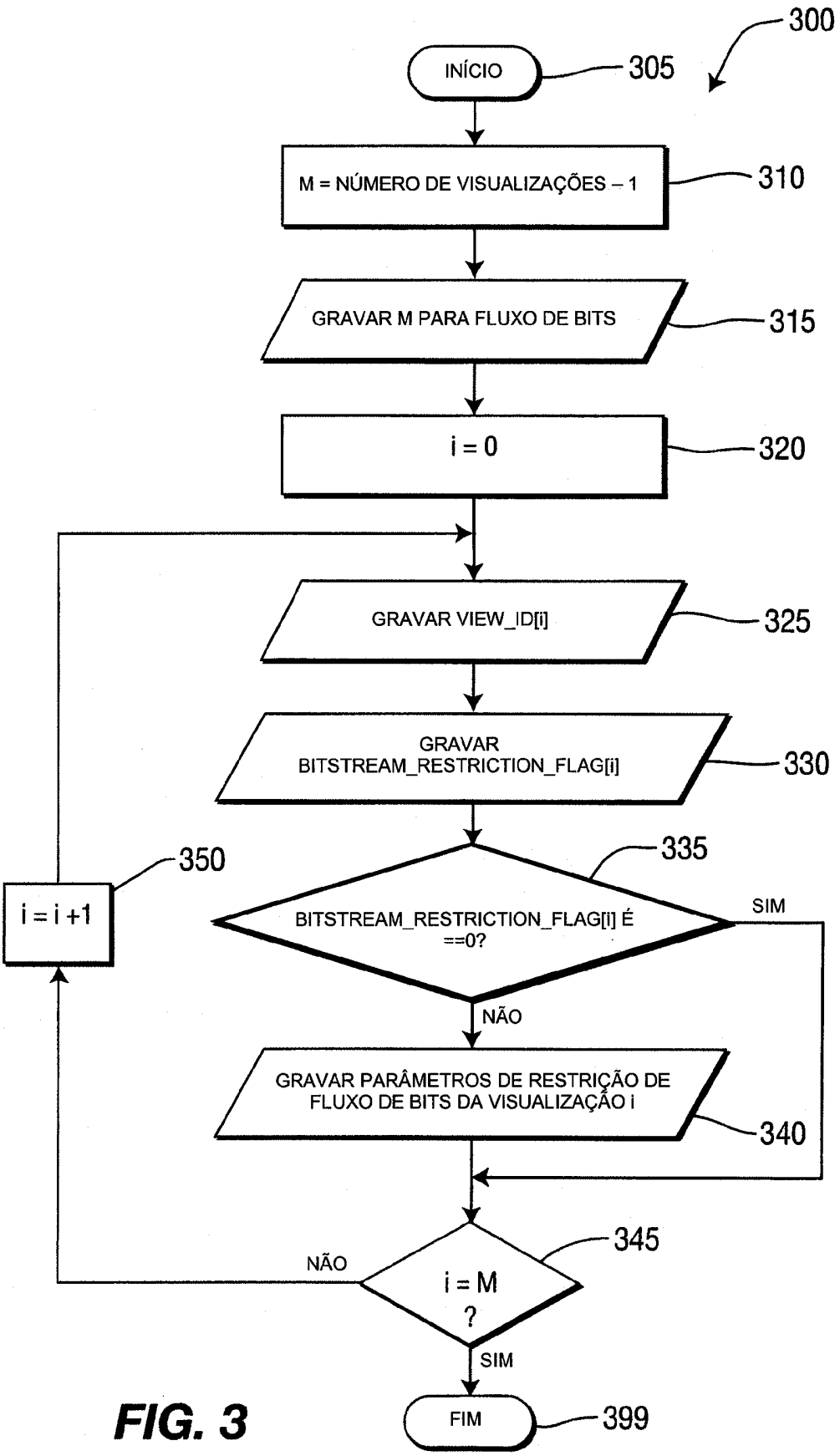

300
INÍCIO
305
M = NÚMERO DE VISUALIZAÇÕES – 1
310
GRAVAR M PARA FLUXO DE BITS
315
i = 0
320
GRAVAR VIEW_ID[i]
325
GRAVAR
BITSTREAM_RESTRICTION_FLAG[i]
330
335
BITSTREAM_RESTRICTION_FLAG[i] É
==0?
SIM
NÃO
350
i = i +1
GRAVAR PARÂMETROS DE RESTRIÇÃO DE
FLUXO DE BITS DA VISUALIZAÇÃO i
340
345
i = M
?
NÃO
SIM
FIM
399

FIG. 3

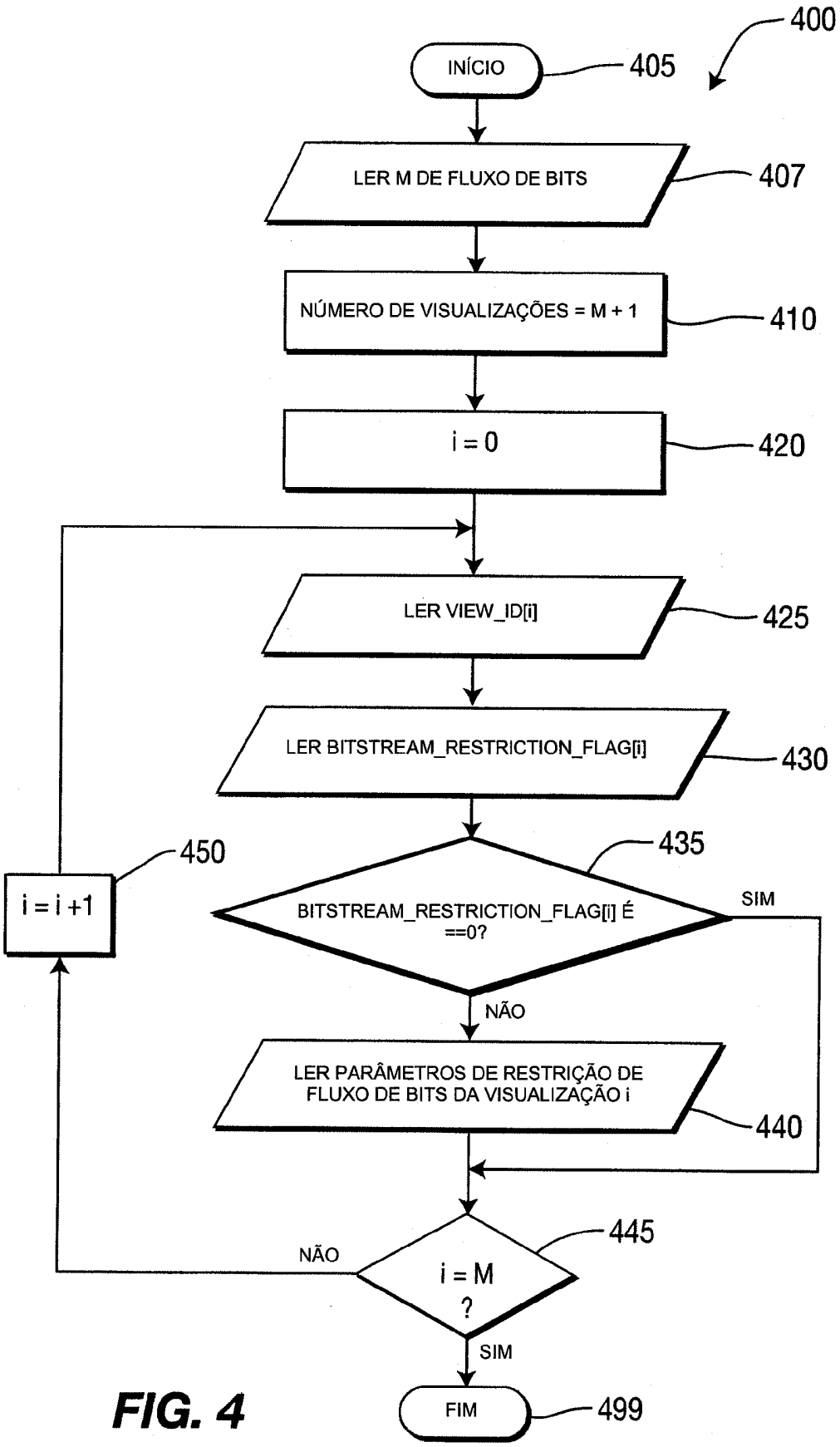

400
INÍCIO
405
LER M DE FLUXO DE BITS
407
NÚMERO DE VISUALIZAÇÕES = M + 1
410
i = 0
420
LER VIEW_ID[i]
425
LER BITSTREAM_RESTRICTION_FLAG[i]
430
435
BITSTREAM_RESTRICTION_FLAG[i] É ==0?
SIM
NÃO
450
i = i +1
LER PARÂMETROS DE RESTRIÇÃO DE FLUXO DE BITS DA VISUALIZAÇÃO i
440
445
i = M ?
NÃO
SIM
FIM
499

FIG. 4

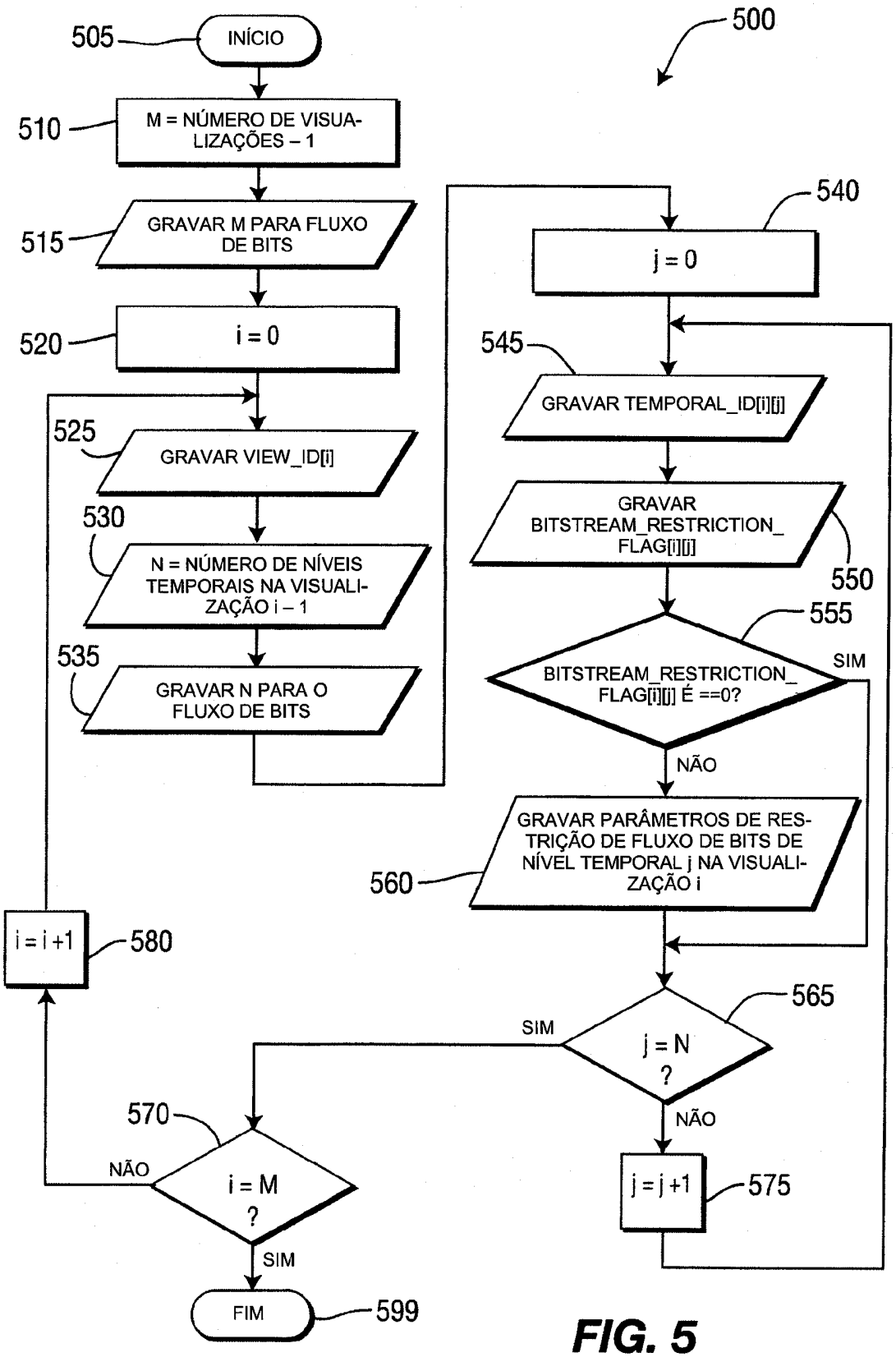

500
505
INÍCIO
510
M = NÚMERO DE VISUALIZAÇÕES – 1
515
GRAVAR M PARA FLUXO DE BITS
520
i = 0
525
GRAVAR VIEW_ID[i]
530
N = NÚMERO DE NÍVEIS TEMPORAIS NA VISUALIZAÇÃO i – 1
535
GRAVAR N PARA O FLUXO DE BITS
540
j = 0
545
GRAVAR TEMPORAL_ID[i][j]
550
GRAVAR BITSTREAM_RESTRICTION_FLAG[i][j]
555
BITSTREAM_RESTRICTION_FLAG[i][j] É ==0?
SIM
NÃO
560
GRAVAR PARÂMETROS DE RESTRIÇÃO DE FLUXO DE BITS DE NÍVEL TEMPORAL j NA VISUALIZAÇÃO i
565
j = N ?
SIM
NÃO
570
i = M ?
NÃO
SIM
575
j = j +1
580
i = i +1
599
FIM

FIG. 5

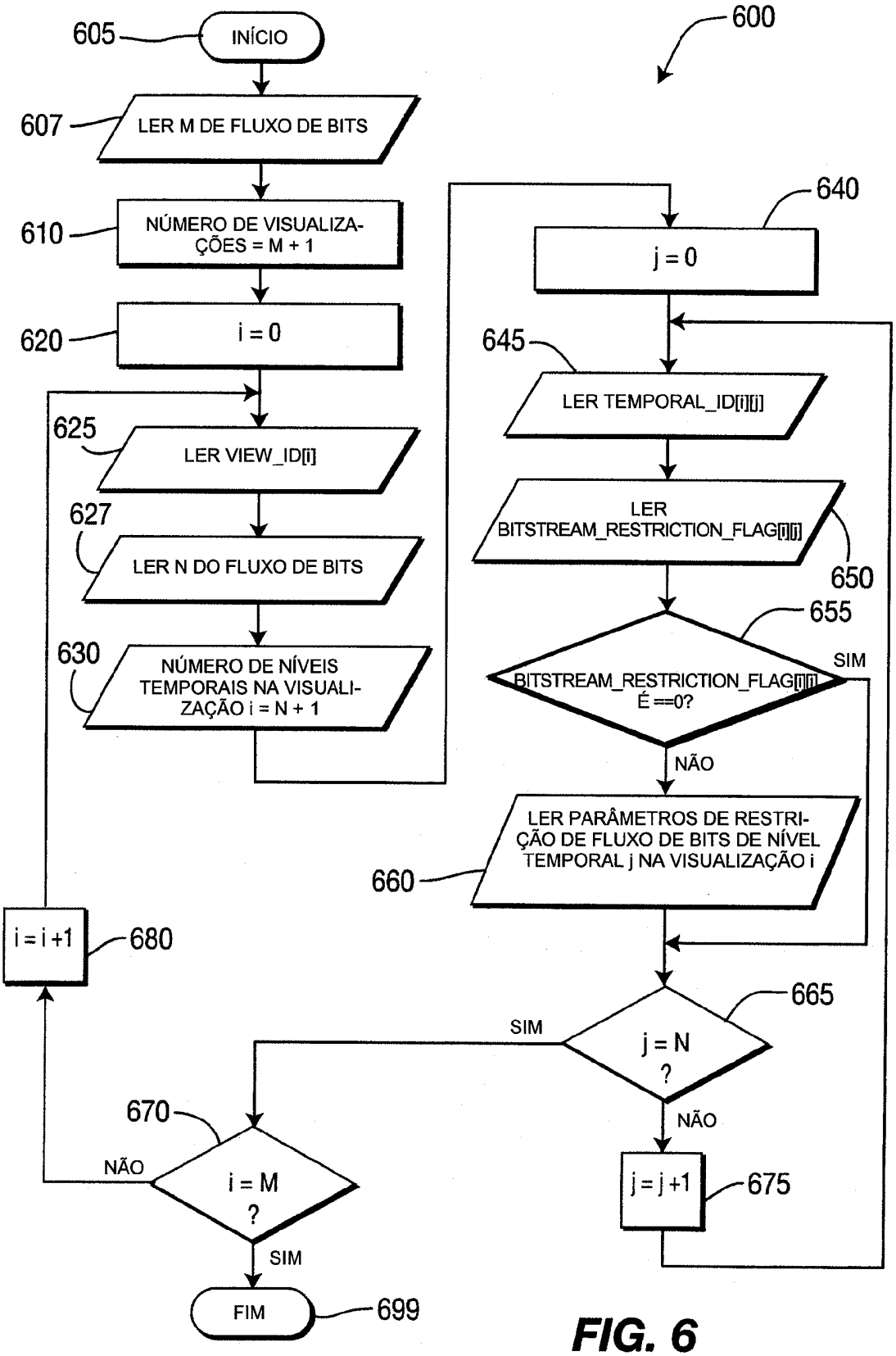

600
605
INÍCIO
607
LER M DE FLUXO DE BITS
610
NÚMERO DE VISUALIZA-ÇÕES = M + 1
620
i = 0
625
LER VIEW_ID[i]
627
LER N DO FLUXO DE BITS
630
NÚMERO DE NÍVEIS TEMPORAIS NA VISUALI-ZAÇÃO i = N + 1
640
j = 0
645
LER TEMPORAL_ID[i][j]
LER BITSTREAM_RESTRICTION_FLAG[i][j]
650
655
BITSTREAM_RESTRICTION_FLAG[i][j] É ==0?
SIM
NÃO
660
LER PARÂMETROS DE RESTRI-ÇÃO DE FLUXO DE BITS DE NÍVEL TEMPORAL j NA VISUALIZAÇÃO i
665
j = N ?
SIM
NÃO
670
i = M ?
NÃO
SIM
675
j = j +1
680
i = i +1
699
FIM

FIG. 6

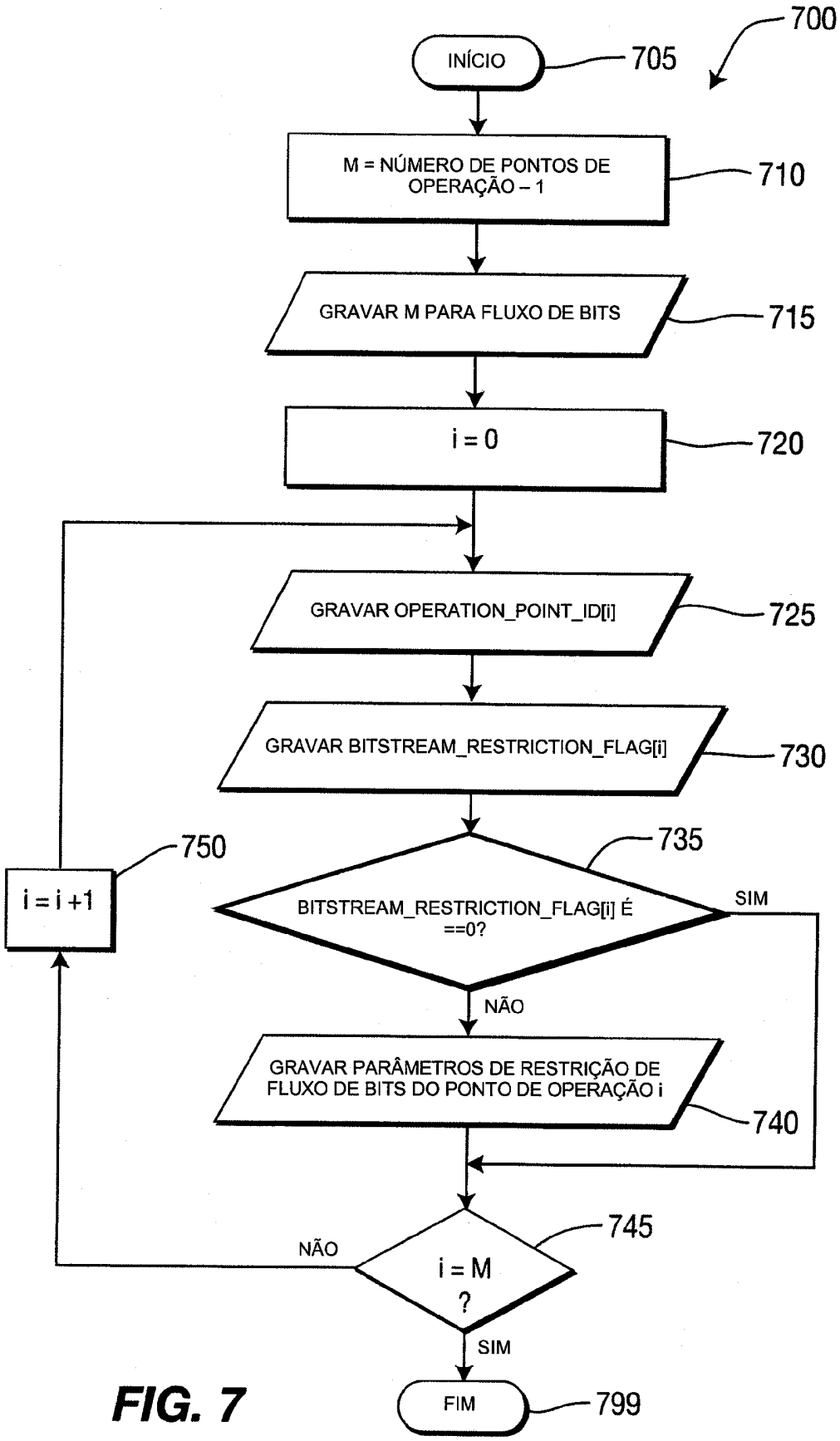

700
INÍCIO
705
M = NÚMERO DE PONTOS DE OPERAÇÃO – 1
710
GRAVAR M PARA FLUXO DE BITS
715
i = 0
720
GRAVAR OPERATION_POINT_ID[i]
725
GRAVAR BITSTREAM_RESTRICTION_FLAG[i]
730
735
BITSTREAM_RESTRICTION_FLAG[i] É ==0?
SIM
NÃO
750
i = i +1
GRAVAR PARÂMETROS DE RESTRIÇÃO DE FLUXO DE BITS DO PONTO DE OPERAÇÃO i
740
745
NÃO
i = M ?
SIM
FIM
799

FIG. 7

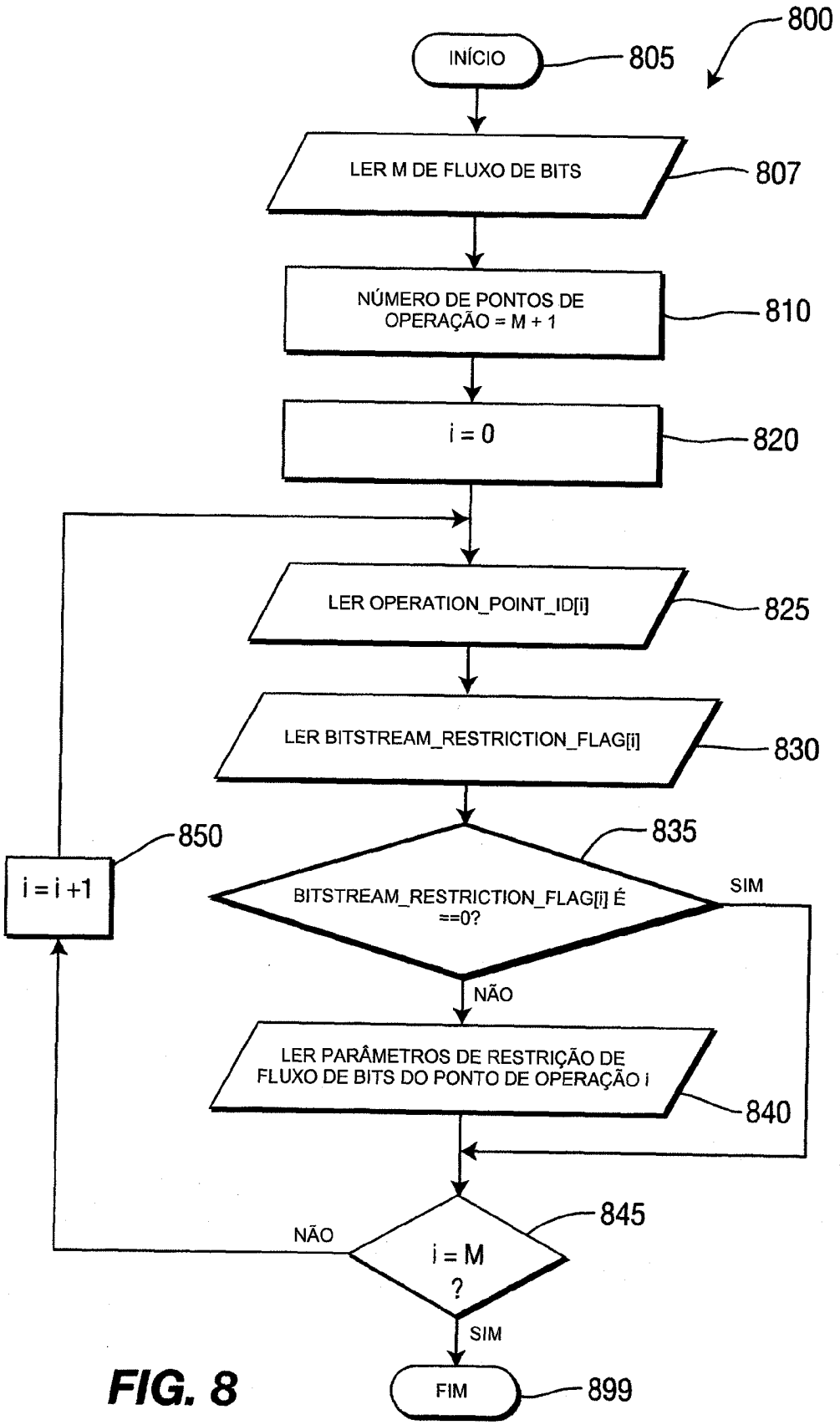

800
INÍCIO
805
LER M DE FLUXO DE BITS
807
NÚMERO DE PONTOS DE OPERAÇÃO = M + 1
810
i = 0
820
LER OPERATION_POINT_ID[i]
825
LER BITSTREAM_RESTRICTION_FLAG[i]
830
835
BITSTREAM_RESTRICTION_FLAG[i] É ==0?
SIM
NÃO
850
i = i +1
LER PARÂMETROS DE RESTRIÇÃO DE FLUXO DE BITS DO PONTO DE OPERAÇÃO i
840
845
NÃO
i = M ?
SIM
FIM
899

FIG. 8

RESUMO

"MÉTODO PARA INCORPORAR INFORMAÇÃO DE USABILIDADE DE VÍDEO (VUI) EM UM SISTEMA DE CODIFICAÇÃO DE VÍDEO DE MÚLTIPLAS VISUALIZAÇÕES (MVC)"

São fornecidos métodos e aparelho para incorporar Informação de Usabilidade de Vídeo (VUI) em codificação de vídeo de múltiplas visualizações (MVC). Um aparelho (100) inclui um codificador (100) para codificar conteúdo de vídeo de múltiplas visualizações ao especificar Informação de Usabilidade de Vídeo para pelo menos um selecionado de: visualizações individuais (300), níveis temporais individuais em uma visualização (500) e pontos de operação individuais (700). Adicionalmente, um aparelho (200) inclui um decodificador para decodificar conteúdo de vídeo de múltiplas visualizações ao especificar Informação de Usabilidade de Vídeo para pelo menos um selecionado de: visualizações individuais (400), níveis temporais individuais em uma visualização (600) e pontos de operação individuais (800).