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

(21) **BR 12 2012 021947-9 A2**

* B R 1 2 2 0 1 2 0 2 1 9 4 7 A 2 *

(22) Data de Depósito: 16/09/2008

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

(54) Título: APARELHO PARA INCORPORAÇÃO DE INFORMAÇÃO DE USABILIDADE DE VÍDEO (VUI) EM UM SISTEMA DE CODIFICAÇÃO DE VÍDEO COM MÚLTIPLOS VISUALIZAÇÕES (MVC)

(51) Int.Cl.: H04N19/46

(52) CPC: H04N19/46

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

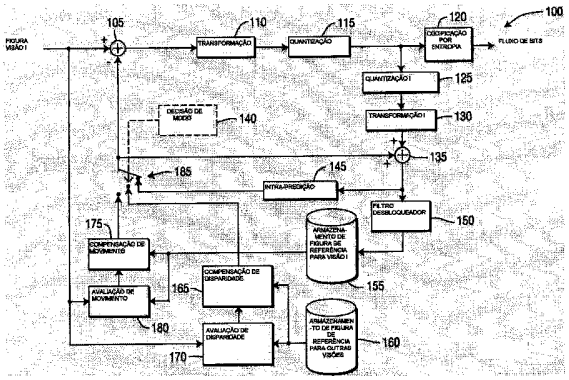
(73) Titular(es): Thomson Licensing

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

(86) Pedido Internacional: PCT US2008010796 de 16/09/2008

(87) Publicação Internacional: WO 2009/048503 de 16/04/2009

(57) Resumo: APARELHO PARA INCORPORAÇÃO DE INFORMAÇÃO DE USABILIDADE DE VÍDEO (VUI) EM UM SISTEMA DE CODIFICAÇÃO DE VÍDEO COM MÚLTIPLOS VISUALIZAÇÕES (MVC) São fornecidos métodos e aparelhos para incorporação de informação de usabilidade de vídeo (VUI) na Codificação de Vídeo com Múltiplas Visualizações (MVC). Um aparelho (100) inclui um codificador (100) para codificação de conteúdo de vídeo com mútiplas visualizações através da especificação e informação de usabilidade de vídeo para pelo menos um selecionado de : visualizações individuais (300) , níveis temporais individuais em uma visualização (500) e pontos operacionais individuais (700). Adicionalmente , um aparelho (200) inclui um decodificador para decodificador de conteúdo de vídeo com múltiplas visualizações através da especificação de informação de usabilidade de vídeo para pelo menos um selecionado de : visualizações individuais (400), níveis temporais individuais em uma visualização (600) e pontos operacionais individuais (800).



"MÉTODO PARA INCORPORAÇÃO DE INFORMAÇÃO DE USABILIDADE DE VÍDEO (VUI) EM UM SISTEMA DE CODIFICAÇÃO DE VÍDEO COM MÚLTIPLAS VISUALIZAÇÕES (MVC)" – **"Dividido do PI 0817508-0 depositado em 16/09/2008".**

Referência Cruzada aos Pedidos Relacionados

Este pedido reivindica o benefício do pedido provisório sob o número de série US60/977.709, depositado em 5 de outubro de 2007, que está incorporado no presente documento em sua totalidade a título de referência. Adicionalmente, este pedido está relacionado ao pedido não-provisório, nº de dossiê PU080155, intitulado "METHODS AND APPARATUS FOR INCORPORATING VIDEO USABILITY INFORMATION (VUI) WITHIN A MULTI- VIEW VIDEO (MVC) CODING SYSTEM", que também reivindica o benefício do pedido provisório sob o número de série US60/977.709, depositado em 5 de outubro de 2007, e que é cedido à mesma requerente, incorporado a título de referência no presente documento, e concomitantemente depositado com o presente.

Campo da Técnica

Os presentes princípios se referem geralmente à codificação e à decodificação de vídeo e, mais particularmente, aos métodos e ao aparelho para incorporação de informação de usabilidade de vídeo (VUI) em uma Codificação de Vídeo com Múltiplas Visualizações (MVC).

Antecedentes

A recomendação H.264 do setor de telecomunicação da União Internacional de Telecomunicações/padrão de Codificação de Vídeo Avançada (AVC) Parte 10 do Grupo de Especialistas em Imagens em Movimento 4 (MPEG-4) da Organização Internacional para Padronização / Comissão Internacional de Eletrotécnica (ISO/IEC) (doravante "padrão MPEG-4 AVC") especifica a sintaxe e a semântica de parâmetros de informação de usabilidade de vídeo (VUI) de conjuntos de parâmetro de sequência. Uma informação de usabilidade de vídeo inclui informações sobre razão de aspecto, varredura em excesso, tipo do sinal de vídeo, localização de croma, temporização, parâmetros do decodificador de referência hipotética (HRD) de camada de abstração de rede (NAL), parâmetros do decodificador de referência hipotética (HRD) de camada de codificação de vídeo (VCL), restrição do fluxo de bits e outros. A informação de usabilidade de vídeo fornece informações extra para um fluxo de bits correspondente para permitir uma aplicação mais ampla para um usuário. Por exemplo, na informação de restrição do fluxo de bits, a informação de usabilidade de vídeo especifica: (1) se o movimento é sobre um delimitador de figura; (2) os bits máximos por figura; (3) os bits máximos por macrobloco; (4) o comprimento máximo do vetor de movimento (horizontal e vertical); (5) o número de quadros de re-ordenação; e (6) o tamanho máximo da memória temporária do quadro decodificado. Quando o decodificador observa a informação, em vez de usar a informação do "nível" para ajustar o requisito de decodifica-

ção, que em general é superior ao que o fluxo de bits requer realmente, o decodificador pode customize sua operação de decodificação com base no limite mais curto.

A Codificação de Vídeo com Múltiplas Visualizações (MVC) é uma extensão para o Padrão MPEG-4 AVC. Na Codificação de Vídeo com Múltiplas Visualizações, as imagens de vídeo para múltiplas visualizações podem ser codificadas através da exploração de uma correlação entre as visualizações. Dentre todas as visualizações, uma visualização é a visualização base, que é compatível com o padrão MPEG-4 AVC e não pode ser predita a partir de outras visualizações. As outras visualizações são chamadas de visualizações não-base. As visualizações não-base podem ser codificadas preditivamente a partir da visualização base e de outras visualizações não-base. Cada visualização pode ser temporariamente subamostrada. Um subconjunto temporal de uma visualização pode ser identificado por um elemento de sintaxe tempora_id. Um nível temporal de uma visualização é uma representação do sinal de vídeo. Pode haver diferentes combinações de visualizações e níveis temporais em um fluxo de bits codificado de vídeo com múltiplas visualizações. Cada combinação é chamada de ponto de operação. Os sub-fluxos de bits, correspondentes aos pontos de operação, podem ser extraídos do fluxo de bits.

Sumário

Estes e outros defeitos e desvantagens da técnica anterior são abordados pelos presentes princípios, que são direcionados aos métodos e ao aparelho para incorporação de informação de usabilidade de vídeo (VUI) na Codificação de Vídeo com Múltiplas Visualizações (MVC).

De acordo com um aspecto dos presentes princípios, é fornecido um aparelho. O aparelho inclui um codificador para codificação de conteúdo de vídeo com múltiplas visualizações através da especificação de informação de usabilidade de vídeo para pelo menos um dentre as visualizações individuais, níveis temporais individuais em uma visualização, e pontos operacionais individuais.

De acordo com outro aspecto dos presentes princípios, é fornecido um método. O método inclui a codificação de conteúdo de vídeo com múltiplas visualizações através da especificação informação de usabilidade de vídeo para pelo menos um dentre visualizações individuais, níveis temporais individuais em uma visualização e pontos operacionais individuais.

De acordo com ainda outro aspecto dos presentes princípios, é fornecido um aparelho. O aparelho inclui um decodificador para decodificação de conteúdo de vídeo com múltiplas visualizações através da especificação informação de usabilidade de vídeo para pelo menos um dentre visualizações individuais, níveis temporais individuais em uma visualização e pontos operacionais individuais.

De acordo com um outro aspecto dos presentes princípios, é fornecido um método.

O método inclui a decodificação conteúdo de vídeo com múltiplas visualizações através da especificação informação de usabilidade de vídeo para pelo menos um dentre visualizações individuais, níveis temporais individuais em uma visualização e pontos operacionais individuais.

Estes e outros aspectos, recursos e vantagens dos presentes princípios se tornarão evidentes a partir da descrição detalhada de modalidades exemplificativas, que deve ser lida juntamente com os desenhos em anexo.

Breve Descrição das Figuras

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

A Figura 1 é um diagrama de bloco de um codificador para Codificação de Vídeo com Múltiplas Visualizações (MVC) exemplificativo, no qual os presentes princípios podem ser aplicados, de acordo com uma modalidade dos presentes princípios;

A Figura 2 é um diagrama de bloco para for para um decodificador de Codificação de Vídeo com Múltiplas Visualizações (MVC) exemplificativo, no qual os presentes princípios podem ser aplicados, de acordo com uma modalidade dos presentes princípios;

A Figura 3 é um diagrama de fluxo para um método exemplificativo para codificação de parâmetros de restrição do fluxo de bits para cada visualização, com o uso de um elemento de sintaxe mvc_vui_parameters_extension(), de acordo com uma modalidade dos presentes princípios;

A Figura 4 é um diagrama de fluxo para um método exemplificativo para decodificação de parâmetros de restrição do fluxo de bits para cada visualização, com um uso de um elemento de sintaxe mvc_vui_parameters_extension(), de acordo com uma modalidade dos presentes princípios;

A Figura 5 é um diagrama de fluxo para um método exemplificativo para codificação de parâmetros de restrição do fluxo de bits para cada nível temporal em cada visualização, com o uso de um elemento de sintaxe mvc_vui_parameters_extension(), de acordo com uma modalidade dos presentes princípios;

A Figura 6 é um diagrama de fluxo para um método exemplificativo para decodificação de parâmetros de restrição do fluxo de bits para cada nível temporal em cada visualização, com o uso de um elemento de sintaxe mvc_vui_parameters_extension(), de acordo com uma modalidade dos presentes princípios;

A Figura 7 é um diagrama de fluxo para um método exemplificativo para codificação de parâmetros de restrição do fluxo de bits para cada ponto de operação, com o uso de um elemento de sintaxe view_scalability_parameters_extension(), de acordo com uma modalidade dos presentes princípios; e

A Figura 8 é um diagrama de fluxo para um método exemplificativo para decodifica-

ção de parâmetros de restrição do fluxo de bits para cada ponto de operação, com o uso de um elemento de sintaxe view_scalability_parameters_extension(), de acordo com uma modalidade dos presentes princípios.

Descrição Detalhada

Os presentes princípios são direcionados aos métodos e ao aparelho para incorporação de informação de usabilidade de vídeo (VUI) na Codificação de Vídeo com Múltiplas Visualizações (MVC).

A presente descrição ilustra os presentes princípios. Dessa forma, será observado que os versados na técnica serão capazes de desenvolver várias disposições que, embora não sejam descritas ou mostradas explicitamente no presente documento, incorporam os presentes princípios e estão incluídas em seu espírito e escopo.

Todos os exemplos e a linguagem condicional mencionados no presente documento são destinados a propósitos pedagógicos para auxiliar o leitor na compreensão dos presentes princípios e dos conceitos cedidos pelo(s) inventor(es) para promoção da técnica, e devem ser constituídos sem limitação a tais exemplos e condições especificamente mencionados.

Além disso, todas as determinações do presente documento que mencionam princípios, aspectos e modalidades dos presentes princípios, bem como exemplos específicos dos mesmos, se destinam a abranger equivalentes tanto estruturais como funcionais dos mesmos. Adicionalmente, pretende-se que tais equivalentes incluam tanto equivalentes conhecidos atualmente quanto os equivalentes desenvolvidos no futuro, isto é, quaisquer elementos desenvolvidos que executam a mesma função, sem levar em consideração a estrutura.

Dessa forma, por exemplo, será observado pelos versados na técnica que os diagramas de bloco apresentados no presente documento representam visualizações conceituais de conjunto de circuitos que incorporam os presentes princípios. Similarmente, será observado que quaisquer fluxogramas, diagramas de fluxo, diagramas de transição de estado, pseudocódigo e similares representam vários processos que podem ser substancialmente representados em meios legíveis por computador e executados assim por um computador ou processador, se ou não tal computador ou processador for explicitamente mostrado.

As funções dos vários elementos mostrados nas figuras podem ser fornecidas através do uso de hardware dedicado bem como hardware capaz de executar software em associação com o 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 destes podem ser compartilhados. Além disso, o uso explícito do termo "processador" ou "controlador" não deve ser construído para se referir exclusivamente a hardware capaz de executar software e

pode incluir implicitamente, sem limitação, hardware de processador de sinal digital ("DSP"), memória de somente leitura ("ROM") para armazenar software, memória de acesso aleatório ("RAM") e armazenamento não-volátil.

Outro hardware, convencional e/ou personalizado, também pode ser incluído. Similarmente, quaisquer comutações nas figuras são apenas conceituais. Sua função pode ser executada através da operação de lógica de programação, através de lógica dedicada, através da interação de controle de programa e lógica dedicada ou ainda manualmente, a técnica particular é selecionável pelo executante conforme mais especificamente compreendido a partir do contexto.

Nas reivindicações deste documento, qualquer elemento denotado como um meio para executar uma função específica é destinado a abranger qualquer maneira de executar aquela função incluindo, por exemplo, a) uma combinação de elementos de circuito que executa aquela função ou b) software em qualquer forma, incluindo, portanto, firmware, microcódigo ou similares, combinados com conjunto de circuitos apropriado para executar aquele software a fim executar a função. Os presentes princípios conforme definido por tais reivindicações estão de acordo com o fato de que as funcionalidades fornecidas pelos vários meios mencionados são combinadas e unidas de maneiras exigidas pelas reivindicações. Isto está, dessa forma, relacionado ao fato de que quaisquer meios que podem fornecer aquelas funcionalidades são equivalentes àqueles mostrados no presente documento.

A referência no relatório descritivo a "aquela modalidade" ou "uma modalidade" dos presentes princípios significa que um recurso, estrutura, características particulares e outros descritos juntamente com a modalidade estão incluídos em pelo menos uma modalidade dos presentes princípios. Dessa forma, as aparências da frase "naquela modalidade" ou "em uma modalidade" aparecem em vários locais por todo o relatório descritivo não se referem necessariamente à mesma modalidade. Deve ser observado que o uso dos termos "e/ou" e "pelo menos um dentre", por exemplo, nos cases de "A e/ou B" e "pelo menos um dentre A e B", é destinado a abranger a seleção da primeira opção mencionada (A) somente, a seleção da segunda opção mencionada (B) somente 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 dentre A, B e C", tal estrutura de frase se destina a abranger a seleção da primeira opção mencionada (A) somente, a seleção da segunda opção mencionada (B) somente, a seleção da terceira opção mencionada (C) somente, a seleção da primeira e da segunda opções mencionadas (A e B) somente, a seleção da primeira e da terceira opções (A e C) somente, da seleção da segunda e da terceira opções mencionadas (B e C) somente ou a seleção de todas as três opções (A, B e C). Isto pode ser estendido, conforme for prontamente evidente para um indivíduo versado nesta técnica e em técnicas relacionadas, para quantos itens forem mencionados.

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

Conforme usado intercambiavelmente no presente documento, "visualização transversal" e "inter-visualização" se referem a figuras que pertencem a uma visualização diferente da visualização atual.

Além disso, conforme usado no presente 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, conforme usado no presente documento, pode se referir, mas não estar limitada, à sintaxe no nível de cabeçalho de fatia, nível de informação de enriquecimento suplementar (SEI), nível de conjunto de parâmetro de figura (PPS) nível de conjunto de parâmetro de sequência (SPS) e nível de cabeçalho de unidade de camada de abstração de rede (NAL). Ademais, deve ficar compreendido que embora uma ou mais modalidades dos presentes princípios sejam descritas no presente documento com propósitos ilustrativos em relação à Codificação de Vídeo com Múltiplas Visualizações extensão do padrão MPEG-4 AVC, os presentes princípios não estão se limitados a somente esta extensão e/ou este padrão e, dessa forma, podem ser utilizados em relação a outros padrões de codificação de vídeo, recomendações e extensões dos mesmos, enquanto é mantido o espírito dos presentes princípios.

Adicionalmente, deve ser observado que embora uma ou mais modalidades dos presentes princípios sejam descritas no presente documento com propósitos ilustrativos em relação à informação de restrição do fluxo de bits, os presentes princípios não se limitam somente ao uso da informação de restrição do fluxo de bits como um tipo de informação de usabilidade de vídeo e, dessa forma, outros tipos de informação de usabilidade de vídeo que podem ser estendidos para uso em relação à codificação de vídeo com múltiplos vídeos também podem ser usados de acordo com os presentes princípios, enquanto é mantido o espírito dos presentes princípios. Voltando à Figura 1, um codificador de Codificação de Vídeo com Múltiplas Visualizações (MVC) exemplificativo é indicado geralmente pelo número de referência 100. O codificador 100 inclui um combinador 105 que tem 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 quantizador 115. Uma saída do quantizador 115 é conectada em comunicação de sinal com uma entrada de um codificador por entropia 120 e uma entrada de um quantizador inverso 125. Uma saída do quantizador 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 de não-inversão de um combinador 135. Uma saída do combinador 135 é conectada em comunicação de sinal com uma entrada de um preditor intra 145 e uma entrada de um filtro desbloqueador 150. Uma saída do filtro desbloqueador 150 é conectada em comunicação de sinal com uma entrada de um armazenamento de figura de referência 155 (para visualização i). Uma saída do armazenamento de figura de referência 155 é conectada em comunicação de sinal com uma primeira entrada de um compensador de movimento 175 e uma primeira entrada de um avaliador de movimento 180. Uma saída do avaliador de movimento 180 é conectada em comunicação de sinal com uma segunda entrada do compensador de movimento 175.

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

Uma saída do decodificador por entropia 120 está disponível como uma saída do codificador 100. Uma entrada de não-inversão do combinador 105 está disponível como uma entrada do codificador 100, e é conectada em comunicação de sinal com uma segunda entrada do avaliador de iluminação/disparidade 170, e uma segunda entrada do avaliador de movimento 180. Uma saída de um comutador 185 é conectada em comunicação de sinal com uma segunda entrada de não-inversão do combinador 135 e com uma entrada de inversão 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 iluminação/disparidade 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 com Múltiplas Visualizações (MVC) exemplificativo é indicado geralmente pelo número de referência 200. O decodificador 200 inclui um decodificador por entropia 205 que tem uma saída conectada em comunicação de sinal com uma entrada de um quantizador inverso 210. Uma saída do quantizador 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 de não-inversão de um combinador 220. Uma saída do combinador 220 é conectada em comunicação de sinal com uma entrada de um filtro desbloqueador 225 e uma entrada de um preditor intra 230. Uma saída do filtro desbloqueador 225 é conectada em comunicação de sinal com uma entrada de um armazenamento de

figura de referência 240 (para visualização i). Uma saída do armazenamento de figura 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 figura de referência 245 (para outras visualizações) é conectada em comunicação de sinal com uma primeira entrada de um compensador de iluminação/disparidade 250. Uma entrada do decodificador por 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 a sintaxe de controle para controlar qual entrada é selecionada pelo comutador 55. Adicionalmente, uma segunda entrada do compensador de movimento 235 está disponível como uma entrada do decodificador 200, para receber vetores de movimento. Ademais, uma segunda entrada do compensador de iluminação/disparidade 250 está disponível como uma entrada para o decodificador 200, para receber sintaxe de compensação de iluminação e vetores de disparidade.

Uma saída de um comutador 255 é conectada em comunicação de sinal com uma segunda entrada de não-inversão do combinador 220. Uma primeira entrada do comutador 255 é conectada em comunicação de sinal com uma saída do compensador de iluminação/disparidade 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 desbloqueador 225 está disponível como uma saída do decodificador. No padrão MPEG-4 AVC, os parâmetros de sintaxe e semântica dos conjuntos de parâmetro de sequência são especificados para informação de usabilidade de vídeo (VUI). Isto representa informações adicionais que podem ser inseridas em um fluxo de bits para acentuar a usabilidade do vídeo para uma ampla variedade de propósitos. A informação de usabilidade de vídeo inclui informações sobre razão de aspecto, varredura em excesso, tipo do sinal de vídeo, localização de croma, temporização, parâmetros do decodificador de referência hipotética (HRD) de camada de abstração de rede (NAL), parâmetros do decodificador de referência hipotética (HRD) de camada de codificação de vídeo (VCL), restrição do fluxo de bits e outras.

De acordo com uma ou mais modalidades dos presentes princípios, é usada este campo de informação de usabilidade de vídeo existente para novos e diferentes propósitos em relação à técnica anterior e, adicionalmente, estende-se seu uso para a Codificação de Vídeo com Múltiplas Visualizações (MVC). No presente esquema de Codificação de Vídeo com Múltiplas Visualizações, a informação de usabilidade de vídeo é estendida de tal modo que possa ser diferente entre, por exemplo, visualizações diferentes, níveis temporais em

uma visualização diferentes ou pontos de operação diferentes. Dessa forma, de acordo com uma modalidade, é especificada a informação de usabilidade de vídeo de acordo com uma ou mais dentre, mas sem limitação, as seguintes: especificação da informação de usabilidade de vídeo para visualizações individuais; especificação da informação de usabilidade de vídeo para níveis temporais individuais em uma visualização e a especificação da informação de usabilidade de vídeo para pontos de operação individuais separadamente.

No padrão MPEG-4 AVC, um conjunto que inclui a informação de usabilidade de vídeo (VUI) pode ser transmitido em um conjunto de parâmetro de sequência (SPS). De acordo com uma modalidade, o conceito de informação de usabilidade de vídeo é estendido para o uso em um contexto de Codificação de Vídeo com Múltiplas Visualizações (MVC). Vantajosamente, isto permite que a informação de usabilidade de vídeo diferente seja especificada para visualizações diferentes, níveis temporais diferentes em uma visualização ou pontos de operação diferentes em Codificação de Vídeo com Múltiplas Visualizações. Em uma modalidade, é fornecida uma abordagem inovadora em consideração, modificação e uso da informação de restrição do fluxo de bits em informação de usabilidade de vídeo para Codificação de Vídeo com Múltiplas Visualizações.

A informação de restrição do 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 do 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_bytes_per_mb_denom	0	ue(v)
log2_max_mv_length_horizontal	0	ue(v)
log2_max_mv_length_vertical	0	ue(v)
num_recorder_frames	0	ue(v)
max_dec_frame_buffering	0	ue(v)
}		
}		

A semântica dos elementos de sintaxe de informação de restrição do fluxo de bits ocorre da seguinte forma:

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

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

motion_vectors_over_pic_boundaries_flag igual a 0 indica que nenhuma amostra fora dos delimitadores da figura e nenhuma amostra em uma posição de amostra fracionária cujo valor é derivado com o uso de uma ou mais amostras fora dos delimitadores da figura são usadas para inter-predizer qualquer amostra.

motion_vectors_over_pic_boundaries_flag igual a 1 indica que uma ou mais amostras fora dos delimitadores da figura podem ser usadas na inter-predição. 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 igual a 1.

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

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

Dependendo do max_bytes_per_pic_denom, o seguinte se aplica:

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

- De outro modo (max_bytes_per_pic_denom não é igual a 0), a figura codificada deve ser representada na sequência de vídeo codificado por mais dos seguintes números 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 igual a 2. A variável PicSizeInMbs é o número de macroblocos na figura. A variável RawMbBits é derivada como na sub-cláusula 7.4.2.1 do padrão MPEG-4 AVC.

max_bits_per_mb_denom indica o número máximo de bits codificados dos dados de macroblock_layer( ) para qualquer macrobloco em qualquer figura da sequência de vídeo codificado. O valor de max_bits_per_mb_denom deve estar na faixa de 0 a 16, inclusive. Dependendo do max_bits_per_mb_denom, o seguinte se aplica:

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

- De outro modo (max_bits_per_mb_denom não é igual a 0), nenhum macrob-

lock_layer( ) codificado deve ser representado no fluxo de bits por mais dos seguintes números de bits.

(128 + RawMbBits)/max_bits_per_mb_denom

Dependendo do entropy_coding_mode_flag, os bits dos dados de macroblock_layer( ) são contados da seguinte forma:

- Se entropy_coding_mode_flag é 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 em sub-cláusulas 9.3.3.2.2 e 9.3.3.2.3 do padrão MPEG-4 AVC quando o macroblock_layer( ) associado ao macrobloco é analisado.

Quando o max_bits_per_mb_denom não está presente, o valor de max_bits _per_mb_denom deve ser igual a 1.

log2_max_mv_length_horizontal e log2_max_mv_length_vertical indicam valor absoluto máximo de um componente de vetor de movimento vertical e horizontal decodificado, respectivamente, em unidades de amostra de ¼ luma, para todas as figuras na sequência de vídeo codificado. Um valor de n afirma que nenhum valor de um componente de vetor de movimento deve exceder a faixa de -2^n a 2^{n-1}, inclusive, em unidades de deslocamento de amostra de ¼ de luma. O valor de log2_max_mv_length_horizontal deve estar na faixa de 0 a 16, inclusive. O valor de log2_max_mv_length_vertical deve estar na faixa de 0 a 16, inclusive. 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 iguais a 16. Deve ser observado que o valor absoluto máximo de um componente de vetor de movimento vertical ou horizontal decodificado também é constrito pelos limites de perfil e nível conforme especificado no Anexo A do padrão MPEG-4 AVC.

num_reorder_frames indica o número máximo de quadros, complementares aos pares de campo ou campos não-pareados que respectivamente precedem qualquer quadro, par de campo complementar ou campo não-pareado na sequência de vídeo codificado em ordem de decodificação e seguem na ordem de saída. O valor de num_reorder_frames deve estar na faixa de 0 a max_dec_frame_buffering, inclusive. Quando o elemento de sintaxe num_reorder_frames não está presente, o valor de num_reorder_frames deve ser inferido da seguinte forma:

- Se profile_idc é igual a 44, 100, 110, 122 ou 244 e constraint_set3_flag é igual a 1 , então, o valor de num_reorder_frames deve 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 igual a

max_dec_frame_bufferingMaxDpbSize.

max_dec_frame_buffering especifica o tamanho requerido da memória temporária de figura decodificada do decodificador de referência hipotética (DPB) em unidades de memórias temporárias de quadro. Uma sequência de vídeo codificado não deve requerer uma memória temporária de figura decodificada com tamanho maior que memórias temporárias de quadro Max(1 , max_dec_frame_buffering) para permitir a saída de figuras decodificadas nos tempos de saída especificados por dpb_output_delay das mensagens de informação de enriquecimento suplementar (SEI) da temporização de figura. O valor de max_dec_frame_buffering deve estar na faixa de num_ref_frames a MaxDpbSize (conforme especificado na sub-cláusula A.3.1 ou A.3.2 do padrão MPEG-4 AVC), inclusive. Quando o elemento de sintaxe max_dec_frame_buffering não está presente, o valor de max_dec_frame_buffering deve ser inferido da seguinte forma:

- Se profile_idc é igual a 44 ou 244 e constraint_set3_flag é igual a 1, então, o valor de max_dec_frame_buffering deve 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 igual a MaxDpbSize.

Na Codificação de Vídeo com Múltiplas Visualizações, os parâmetros de restrição do fluxo de bits personalizam a operação de decodificação de um sub-fluxo com base nos limites mais curtos. Portanto, deve-se permitir que os parâmetros de restrição do fluxo de bits sejam especificados para cada sub-fluxo extraível de um fluxo de bits codificado de vídeo com múltiplas visualizações. De acordo com uma modalidade, é proposta especificação da informação de restrição do fluxo de bits para cada visualização, para cada nível temporal em uma visualização e/ou para cada ponto de operação.

Especificação dos parâmetros de restrição do fluxo de bits para cada visualização

Os parâmetros de restrição de fluxo de bit podem ser especificados para cada visualização. É proposta a sintaxe de mvc_vui_parameters_extension, que é uma parte do subset_sequence_parameter_set. A Tabela 2 ilustra a sintaxe de mvc_vui_parameters_extension.

mvc_vui_parameters_extension() enlaça todas as visualizações que estão associadas a este subset_sequence_parameter_set. O view_id de cada visualização e os parâmetros de restrição do fluxo de bits de cada visualização são especificados no interior do enlace.

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]		
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_bytes_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)
}		
}		
}		

A semântica da restrição do fluxo de bits elementos de sintaxe é da seguinte forma:

bitstream_restriction_flag[i] especifica o valor de bitstream_restriction_flag da visualização que 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 que tem 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 que tem view_id[i] igual a view_id deve ser igual a 1.

max_bytes_per_pic_denom[i] especifica o valor de max_bytes_per_pic_denom da visualização que tem 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 que tem view_id[i] igual a view_id deve ser igual a 2.

max_bits_per_mb_denom[i] especifica o valor de max_bits_per_mb_denom da visualização que tem view_id[i] igual a view_id. Quando o max_bits_per_mb_denom[i] não está presente, o valor de max_bits_per_mb_denom da visualização que tem view_id[i] igual a view_id deve ser igual a 1.

log2_max_mv_length_horizontal[i] e log2_max_mv_length_vertical[i] respectivamente especificam os valores de log2_max_mv_length_horizontal e log2_max_mv_length_vertical da visualização que tem 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 que tem view_id[i] igual a view_id deve ser igual a 16.

num_reorder_frames[i] especifica o valor de num_reorder_frames da visualização que tem 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, inclusive. Quando o elemento de sintaxe num_reorder_frames[i] não está presente, o valor de num_reorder_frames da visualização que tem view_id[i] igual a view_id deve ser igual a max_dec_frame_buffering.

max_dec_frame_buffering[i] especifica o valor de max_dec_frame_buffering da visualização que tem 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 (conforme especificado na sub-cláusula A.3.1 ou A.3.2 no padrão MPEG-4 AVC), inclusive. Quando o elemento de sintaxe max_dec_frame_buffering[i] não está presente, o valor de max_dec_frame_buffering da visualização que tem view_id[i] igual a view_id deve ser igual a MaxDpbSize.

Voltando à Figura 3, um método exemplificativo para codificação de parâmetros de restrição do fluxo de bits para cada visualização, com o uso de um elemento de sintaxe mvc_vui_parameters_extension(), é indicado geralmente pelo número de referência 300. O método 300 inclui um bloco de início 305 que passa controle para um bloco de função 310. O bloco de função 310 ajusta 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 ajusta 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 ou não o elemento de sintaxe bitstream restriction_flag[i] é igual a zero. Se sim, 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 do fluxo de bits de visualização i, e passa o controle para o bloco de decisão 345. O bloco de decisão 345 determina se ou não a variável i é igual à variável M. Se sim, então, o controle é passado para um bloco de extremidade 399. De outro modo, o controle é passado para um bloco de função 350.

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

Voltando à Figura 4, um método exemplificativo para decodificação de parâmetros de restrição do fluxo de bits para cada visualização, com o uso de um elemento de sintaxe mvc_vui_parameters_extension(), é indicado geralmente 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 ajusta 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 ajusta 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 ou não o elemento de sintaxe bitstream restriction_flag[i] é igual a zero. Se sim, então, o controle é 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 do fluxo de bits de visualização i, e passa o controle para o bloco de decisão 445. O bloco de decisão 445 determina se ou não a variável i é igual à variável M. Se sim, então, o controle é passado para um bloco de extremidade 499. De outro modo, o controle é passado para um bloco de função 450.

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

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

Os parâmetros de restrição de fluxo de bits podem ser especificados para cada nível temporal de cada visualização. É proposta a sintaxe de mvc_vui_parameters_extension, que é uma parte de subset_sequence_parameter_set. A Tabela 3 ilustra a sintaxe de mvc_vui_parameters_extension.

Tabela 3

mvc_mui_parameters_extension(){	C	Descritor
num_views_minus1	0	ue(v)
for(i = 0; i<= num_views_minus1; i++){		
view_id[i]	0	
num_temporal_layers_in_view_minus1[i]	0	
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]	0	ue(v)
max_bytes_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)
}		

}		
}		
}		

A semântica dos elementos de sintaxe da restrição do fluxo de bits é da seguinte forma:

bitstream_restriction_flag [i][j] especifica o valor de bitstream restriction_flag do nível temporal que tem temporal_id[i][j] igual a temporal_id na visualização que tem 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 que tem temporal_id[i][j] igual a temporal_id na visualização que tem 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 que tem temporal_id[i][j] igual a temporal_id na visualização que tem view_id[i] igual a view_id deve ser igual a 1.

max_bytes_per_pic_denom[i][j] especifica o valor de max_bytes_per_pic_denom do nível temporal que tem temporal_id[i][j] igual a temporal_id na visualização que tem 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 que tem temporal_id[i][j] igual a temporal_id na visualização que tem view_id[i] igual a view_id deve ser igual a 2.

max_bits_per_mb_denom [i][j] especifica o valor de max_bits_per_mb_denom valor do nível temporal que tem temporal_id[i][j] igual a temporal_id na visualização que tem view_id[i] igual a view_id. Quando o max_bits_per_mb_denom[i] não está presente, o valor de max_bits_per_mb_denom do nível temporal que tem temporal_id[i][j] igual a temporal_id na visualização que tem view_id[i] igual a view_id deve ser igual a 1.

log2_max_mv_length_horizontal[i][j] e log2_max_mv_length_vertical[i][j] respectivamente especificam os valores de log2_max_mv_length_horizontal e log2_max_mv_length_vertical do nível temporal que tem temporal_id[i][j] igual a temporal_id na visualização que tem 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 que tem temporal_id[i][j] igual a temporal_id na visualização que tem view_id[i] igual a view_id deve ser igual a 16.

num_reorder_frames[i][j] especifica o valor de num_reorder_frames do nível temporal que tem temporal_id[i][j] igual a temporal_id na visualização que tem 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, inclusive. Quando o elemento de sintaxe num_reorder_frames[i] não está presente, o valor de num_reorder_frames do nível temporal que tem temporal_id[i][j] igual a temporal_id na visualização que tem view_id[i] igual a view_id deve ser igual a max_dec_frame_buffering.

max_dec_frame_buffering[i][j] especifica o valor de max_dec_frame_buffering do nível temporal que tem temporal_id[i][j] igual a temporal_id na visualização que tem 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 (conforme especificado na sub-cláusula A.3.1 ou A.3.2 no padrão MPEG-4 AVC), inclusive. Quando o elemento de sintaxe max_dec_frame_buffering[i] não está presente, o valor de max_dec_frame_buffering do nível temporal que tem temporal_id[i][j] igual a temporal_id na visualização que tem view_id[i] igual a view_id deve ser igual a MaxDpbSize.

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

Voltando à Figura 5, um método exemplificativo para codificação de parâmetros de restrição do fluxo de bits para cada nível temporal em cada visualização, com o uso de um elemento de sintaxe mvc_vuLparameters_extension(), é indicado geralmente 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 ajusta 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 ajusta 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 ajusta uma variável N igual a um número de níveis temporais in 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 ajusta 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 ou não o elemento de sintaxe bitstream restriction_flag[i][j] é igual a zero. Se sim, então, controle é 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 do fluxo de bits do nível temporal j na visualização i, e passa o controle para o bloco de decisão 565. O bloco de decisão 565 determina se ou não a variável j é igual à variável N. Se sim, então, o controle é 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 ou não a variável i é igual à variável M. Se sim, então, o controle é passado para um bloco de extremidade 599. De outro modo, o controle é passado para um bloco de função 580.

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

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

Voltando à Figura 6, um método exemplificativo para decodificação de parâmetros de restrição do fluxo de bits para cada nível temporal em cada visualização, com o uso de um elemento de sintaxe mvc_vui_parameters_extension(), é indicado geralmente 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 ajusta um número de visualizações igual a M mais um, e passa o controle para um bloco de função 620. O bloco de função 620 ajusta 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 ajusta 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 ajusta 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 ou não o elemento de sintaxe bitstream restriction_flag[i][j] é igual a zero. Se sim, então, o controle é 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 do fluxo de bits do nível temporal j na visualização i, e passa o controle para o bloco de decisão 665. O bloco de decisão 665 determina se ou não a variável j é igual à variável N. Se sim, então, o controle é 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 ou

não a variável i é igual à variável M. Se sim, então, o controle é passado para um bloco de extremidade 699. De outro modo, o controle é passado para um bloco de função 680.

O bloco de função 680 ajusta a variável i igual a i mais um, e retorna o controle para o bloco de função 625. O bloco de função 675 ajusta a variável j igual a j mais um, e retorna o controle para o bloco de função 645.

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

Os parâmetros de restrição de fluxo de bits podem ser especificados para cada ponto de operação. É proposta condução dos parâmetros de restrição do fluxo de bits de cada ponto de operação na mensagem SEI de informação de escalabilidade de visualização. A sintaxe da mensagem SEI de informação de escalabilidade de visualização pode ser modificada como na Tabela 4. A sintaxe para informação de restrição do fluxo de bits é inserida em um enlace que enlaça 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_parameters_sets_info_present_flag[i]	5	u(1)
bistream_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/* igual a 0*/	5	u(1)

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][j]){		
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_seq_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	ue(v)
max_bytes_per_pic_denom[i]	0	ue(v)
max_bytes_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)
}		
}		
}		

A semântica dos elementos de sintaxe de restrição do fluxo de bits é da seguinte forma:

bitstream_restriction_flag[i] especifica o valor of bitstream restriction_flag do ponto de operação que tem operation_point_id[i] igual a operation_point_id.

motion_vectors_over_pic_boundaries_flag[i] especifica o valor do motion_vectors_over_pic_boundaries_flag do ponto de operação que tem 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 que tem operation_point_id[i] igual a operation_point_id deve ser igual a 1.

max_bytes_per_pic_denom [i] especifica o valor max_bytes_per_pic_denom do ponto de operação que tem 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 que tem operation_point_id[i] igual a operation_point_id deve ser igual a 2.

max_bits_per_mb_denom [i] especifica o valor de max_bits_per_mb_denom do ponto de operação que tem operation_point_id[i] igual a operation_point_id. Quando o max_bits_per_mb_denom[i] não está presente, o valor de max_bits_per_mb_denom do ponto de operação que tem operation_point_id[i] igual a operation_point_id deve ser igual a 1.

log2_max_mv_length_horizontal[i] e log2_max_mv_length_vertical[i] respectivamente especificam o valor de log2_max_mv_length_horizontal e o valor de log2_max_mv_length_vertical do ponto de operação que tem 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 que tem operation_point_id[i] igual a operation_point_id deve ser igual a 16.

num_reorder_frames [i] especifica o valor de num_reorder_frames do ponto de operação que tem 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, inclusive. Quando o elemento de sintaxe num_reorder_frames[i] não está presente, o valor de num_reorder_frames do ponto de operação que tem operation_point_id[i] igual a operati-

on_point_id deve ser igual a max_dec_frame_buffering.

max_dec_frame_buffering [i] especifica o valor de max_dec_frame_buffering do ponto de operação que tem 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 (conforme especificado na sub-cláusula A.3.1 ou A.3.2 no padrão MPEG-4 AVC), inclusive. Quando o elemento de sintaxe max_dec_frame_buffering[i] não está presente, o valor de max_dec_frame_buffering do ponto de operação que tem operation_point_id[i] igual a operation_point_id deve ser igual a MaxDpbSize.

Voltando à Figura 7, um método exemplificativo para codificação de parâmetros de restrição do fluxo de bits para cada ponto de operação, com o uso de um elemento de sintaxe view_scalability_parameters_extension(), é indicado geralmente 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 ajusta 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 ajusta 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 ou não o elemento de sintaxe bitstream restriction_flag_id[i] é igual a zero. Se sim, então, o controle é 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 do 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 ou não a variável i é igual à variável M. Se sim, então, o controle é passado para um bloco de extremidade 799. De outro modo, o controle é passado para um bloco de função 750. O bloco de função 750 ajusta a variável i igual a i mais um, e retorna o controle para o bloco de função 725.

Voltando à Figura 8, um método exemplificativo para decodificação de parâmetros de restrição do fluxo de bits para cada ponto de operação, com o uso de um elemento de sintaxe view_scalability_parameters_extension(), é indicado geralmente 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 a variável M de um fluxo de bits, e passa o controle para um bloco de função 810. O bloco de função 810 ajusta um número de pontos de operação igual a M mais um, e passa o controle para um bloco de função 820. O bloco de fun-

ção 820 ajusta 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 ou não o elemento de sintaxe bitstream restriction_flag[i] é igual a zero. Se sim, então, o controle é 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 do fluxo de bits do ponto de operação i, e passa o controle para o bloco de decisão 845. O bloco de decisão 845 determina se ou não a variável i é igual à variável M. Se sim, então, o controle é passado para um bloco de extremidade 899. De outro modo, o controle é passado para um bloco de função 850.

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

Uma descrição será fornecida no presente momento sobre alguns dos muitos recursos/vantagens acompanhantes da presente invenção, alguns destes são mencionados acima. Por exemplo, uma vantagem/recurso é um aparelho que inclui um codificador para codificação de conteúdo de vídeo com múltiplas visualizações através da especificação de informação de usabilidade de vídeo para pelo menos um dentre visualizações individuais, níveis temporais individuais em uma visualização e pontos operacionais individuais.

Outra vantagem/recurso é o aparelho que tem o codificador conforme descrito acima, sendo que os parâmetros são especificados em pelo menos um elemento de sintaxe de alto nível.

Além disso, outra vantagem/recurso é o aparelho que tem o codificador conforme descrito acima, sendo que o pelo menos um elemento de sintaxe de alto nível inclui pelo menos um dentre um elemento de sintaxe mvc_vui_parameters_extension(), uma mensagem de sintaxe de informação de enriquecimento suplementar mvc_scalability_info, pelo menos uma porção de um conjunto de parâmetros de sequência, um conjunto de parâmetros de figura e informação de enriquecimento suplementar.

Adicionalmente, outra vantagem/recurso é o aparelho que tem o codificador conforme descrito acima, sendo que pelo menos uma porção da informação de usabilidade de vídeo compreende parâmetros de restrição do fluxo de bits.

Estes e outros recursos e vantagens dos presentes princípios podem ser prontamente confirmados pelo versado na técnica pertinente com base nos ensinamentos do presente documento. Deve ficar entendido que os ensinamentos dos presentes princípios podem ser implantados de várias formas de hardware, software, firmware, processadores com propósitos especiais ou combinações destes.

Mais preferencialmente, os ensinamentos dos presentes princípios são implantados como uma combinação de hardware e software. Além disso, o software pode ser implantado como um programa aplicativo incorporado de forma tangível em uma unidade de armazenamento de programa. O programa pode ser atualizado e executado por uma máquina que compreende qualquer arquitetura adequada. Preferencialmente, a máquina é implantada em uma plataforma de computador que tem hardware tal como uma ou mais unidades de processamento central ("CPU"), uma memória de acesso aleatório ("RAM") e interfaces de entrada/saída ("I/O"). A plataforma de computador também pode incluir um sistema operacional e um código de micro-instrução. Os vários processos e funções descritos no presente documento podem ser tanto parte do código de micro-instrução como parte do programa aplicativo, ou qualquer combinação destes, que podem ser executados por um CPU. Além disso, várias outras unidades periféricas podem ser conectadas à plataforma de computador tal como uma unidade de armazenamento de dados adicional e uma unidade de impressão.

Deve ser compreendido adicionalmente que, devido ao fato de alguns dos componentes e métodos do sistema constituinte revelados nos desenhos em anexo serem preferencialmente implantados em software, as conexões reais entre os componentes do sistema ou os blocos de funções do processo podem diferir dependendo da maneira na qual os presentes princípios são programados. Diante dos ensinamentos do presente documento, o versado na técnica pertinente será capaz de contemplar estas implantações ou configurações similares dos presentes princípios.

Embora as modalidades ilustrativas tenham sido descritas no presente documento em referência aos desenhos em anexo, deve ficar compreendido que os presentes princípios não estão limitados àquelas exatas modalidades, e que várias alterações e modificações podem ser efetuadas nisso por um indivíduo versado na técnica pertinente sem que se afaste do escopo ou do espírito dos presentes princípios. Todas as tais alterações e modificações se destinam a estar incluídas no escopo dos presentes princípios conforme descrito nas reivindicações em anexo.

REIVINDICAÇÃO

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

codificar conteúdo de vídeo de múltiplas visualizações através da especificação de informação de usabilidade de vídeo para pontos de operação individuais em que a informação de usabilidade de vídeo compreende um ou mais parâmetros de restrição do fluxo de bits e os parâmetros de restrição do fluxo de bits são incluídos em uma mensagem de informação de enriquecimento suplementar.

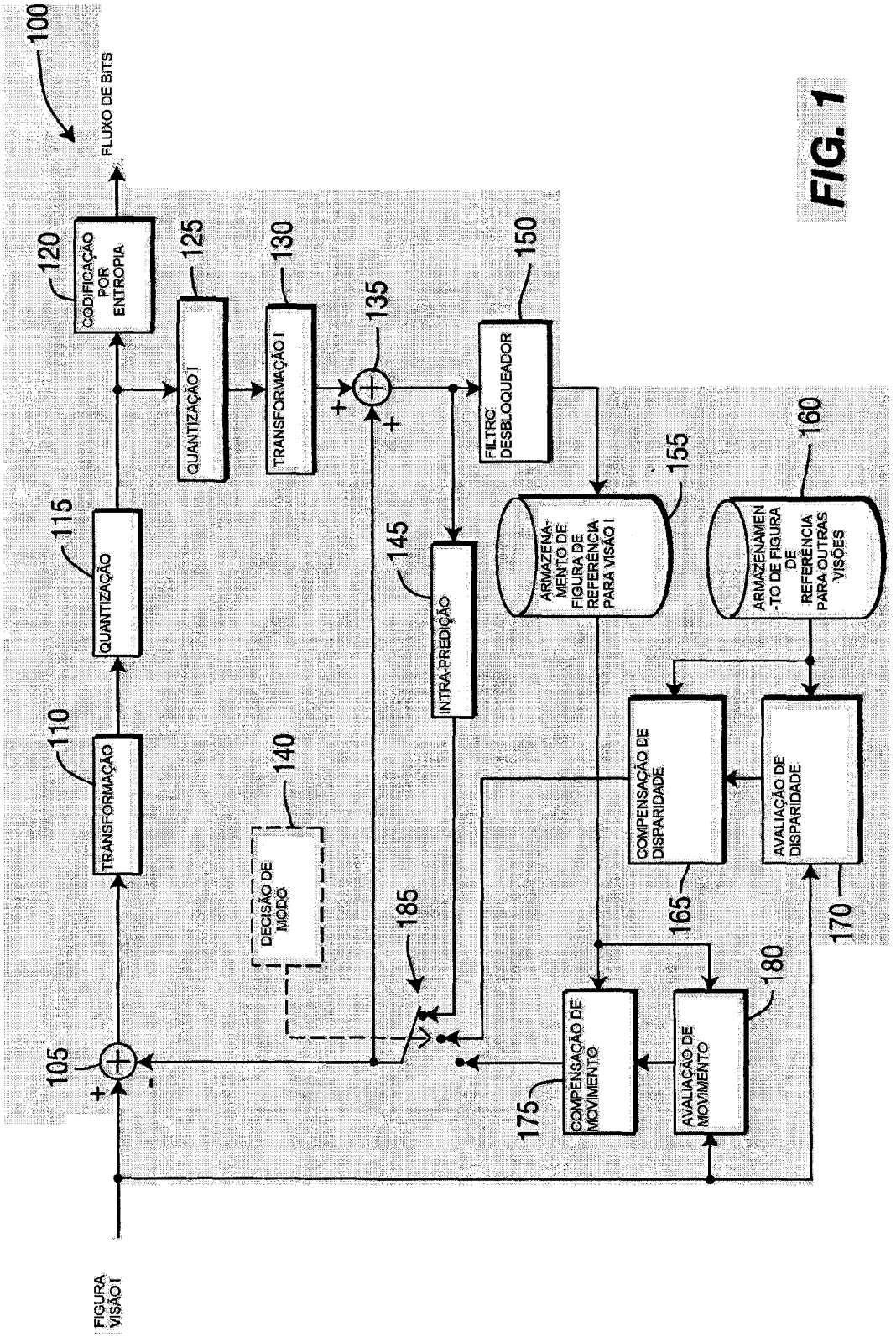

100
FLUXO DE BITS
120
CODIFICAÇÃO POR ENTROPIA
125
QUANTIZAÇÃO I
130
TRANSFORMAÇÃO I
135
150
FILTRO DESBLOQUEADOR
155
ARMAZENAMENTO DE FIGURA DE REFERÊNCIA PARA VISÃO I
160
ARMAZENAMENTO DE FIGURA DE REFERÊNCIA PARA OUTRAS VISÕES
115
QUANTIZAÇÃO
110
TRANSFORMAÇÃO
140
DECISÃO DE MODO
145
INTRA-PREDIÇÃO
185
105
+
-
175
COMPENSAÇÃO DE MOVIMENTO
AVALIAÇÃO DE MOVIMENTO
180
165
COMPENSAÇÃO DE DISPARIDADE
AVALIAÇÃO DE DISPARIDADE
170
FIGURA VISÃO I

FIG. 1

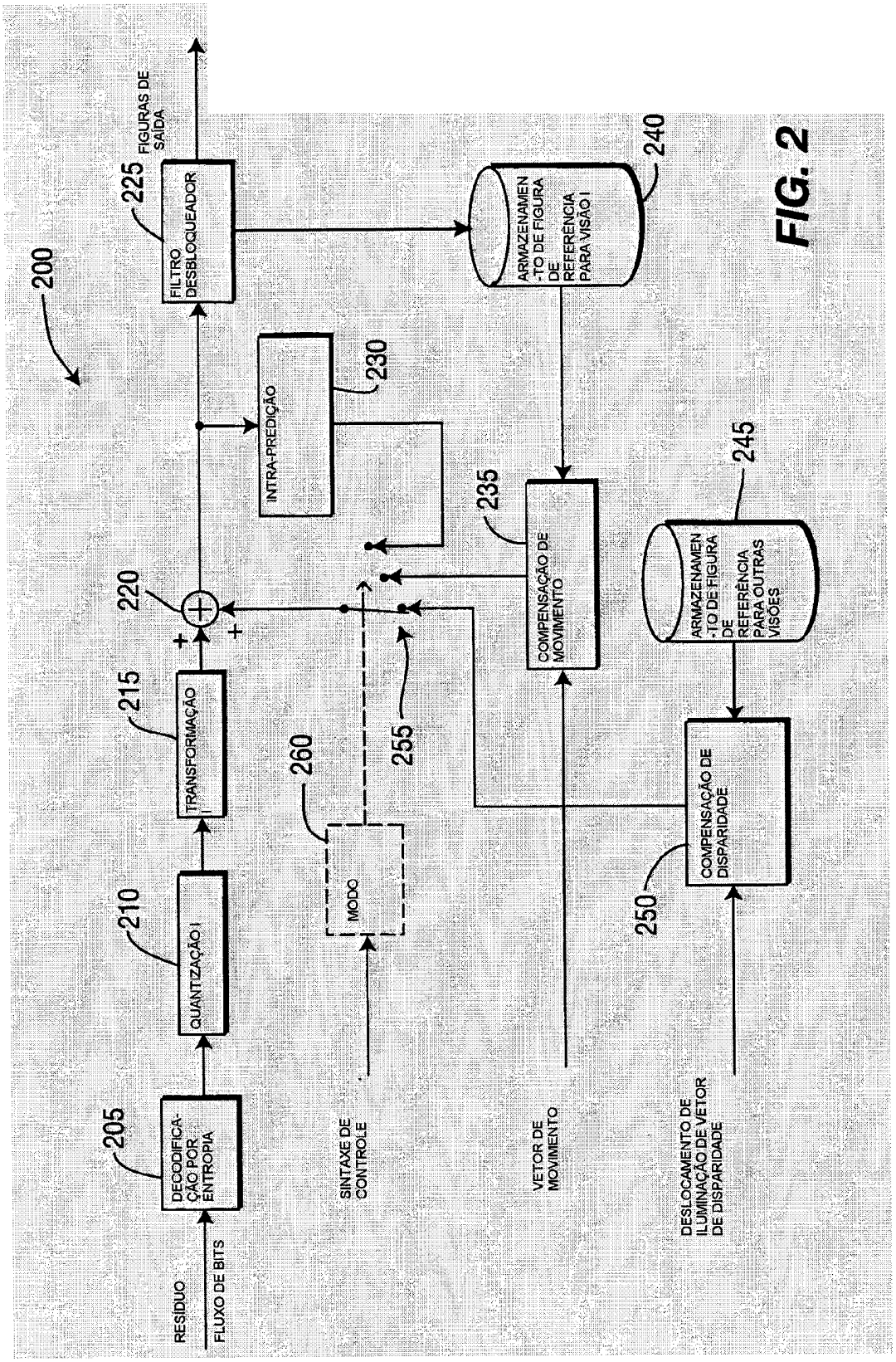

200
205
DECODIFICA-ÇÃO POR ENTROPIA
210
QUANTIZAÇÃO I
215
TRANSFORMAÇÃO I
220
225
FILTRO DESBLOQUEADOR
FIGURAS DE SAÍDA
230
INTRA-PREDIÇÃO
235
COMPENSAÇÃO DE MOVIMENTO
240
ARMAZENAMEN-TO DE FIGURA DE REFERÊNCIA PARA VISÃO I
245
ARMAZENAMEN-TO DE FIGURA DE REFERÊNCIA PARA OUTRAS VISÕES
250
COMPENSAÇÃO DE DISPARIDADE
255
260
MODO
RESÍDUO
FLUXO DE BITS
SINTAXE DE CONTROLE
VETOR DE MOVIMENTO
DESLOCAMENTO DE ILUMINAÇÃO DE VETOR DE DISPARIDADE

FIG. 2

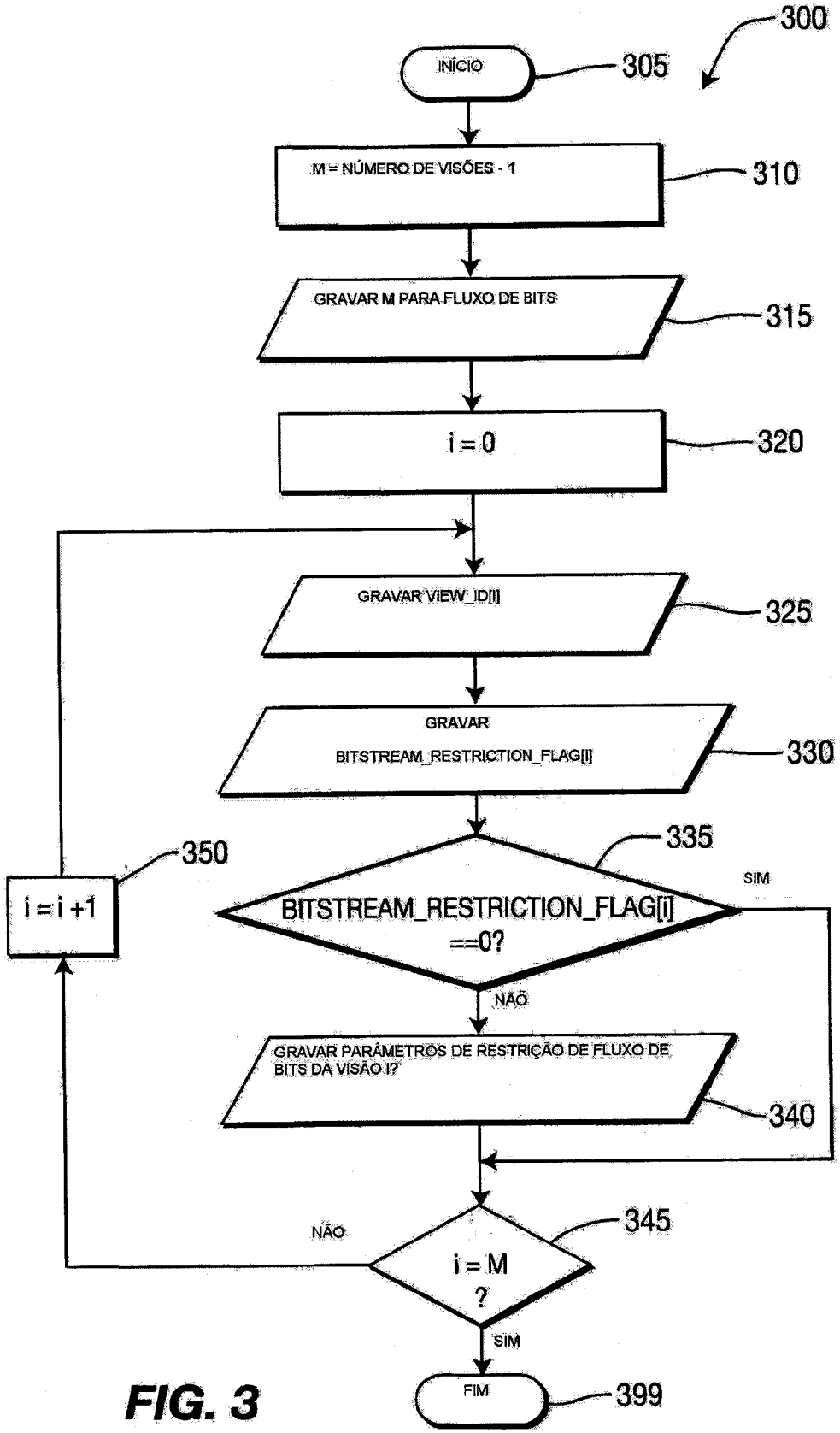

300
INÍCIO
305
M = NÚMERO DE VISÕES - 1
310
GRAVAR M PARA FLUXO DE BITS
315
i = 0
320
GRAVAR VIEW_ID[i]
325
GRAVAR
BITSTREAM_RESTRICTION_FLAG[i]
330
335
BITSTREAM_RESTRICTION_FLAG[i]
==0?
SIM
NÃO
GRAVAR PARÂMETROS DE RESTRIÇÃO DE FLUXO DE BITS DA VISÃO i?
340
350
i = i +1
345
i = M
?
NÃO
SIM
FIM
399

FIG. 3

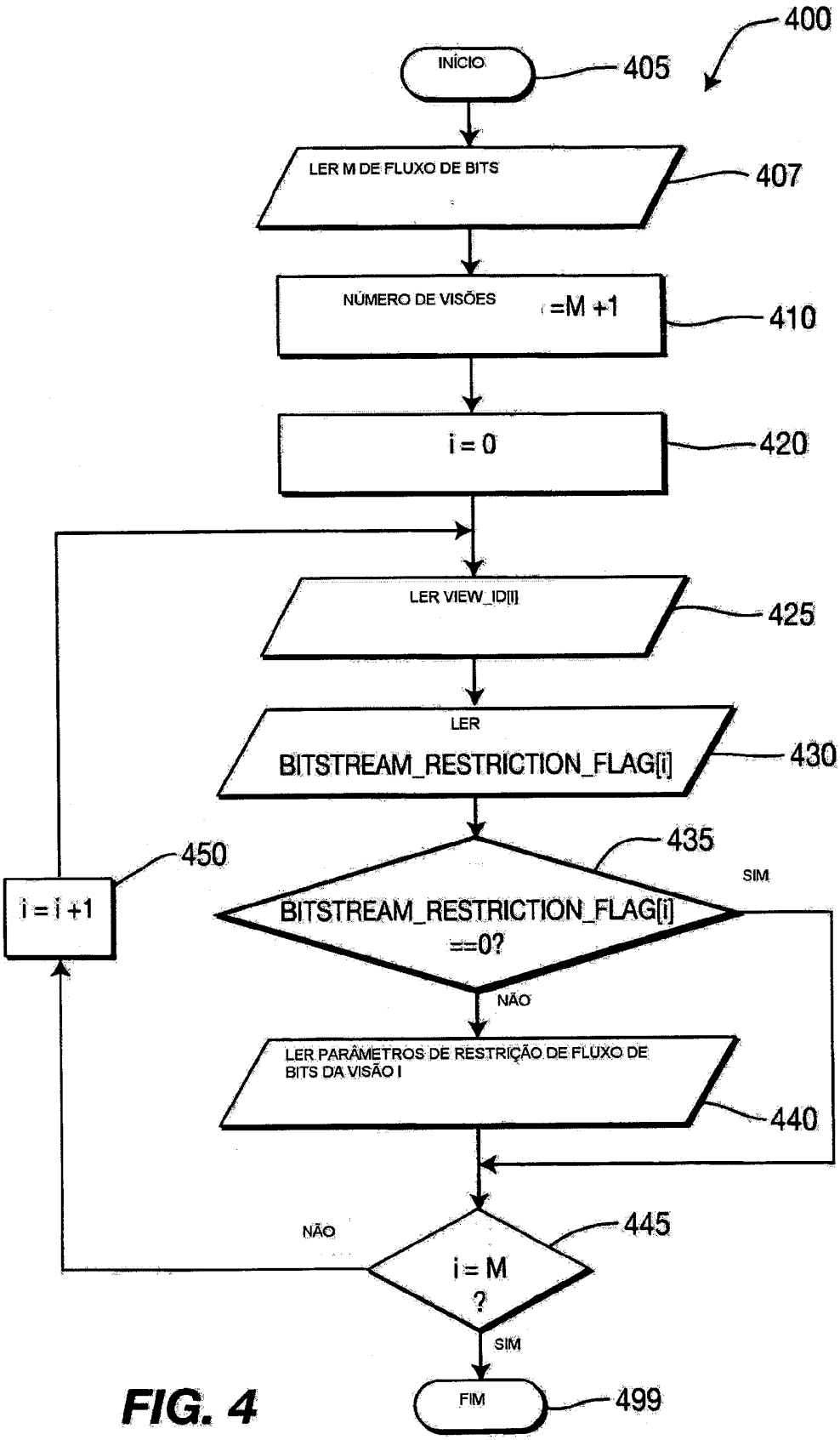

400
INÍCIO
405
LER M DE FLUXO DE BITS
407
NÚMERO DE VISÕES =M +1
410
i = 0
420
LER VIEW_ID[i]
425
LER
BITSTREAM_RESTRICTION_FLAG[i]
430
435
SIM
BITSTREAM_RESTRICTION_FLAG[i] ==0?
NÃO
450
i = i +1
LER PARÂMETROS DE RESTRIÇÃO DE FLUXO DE BITS DA VISÃO I
440
NÃO
445
i = M ?
SIM
FIM
499

FIG. 4

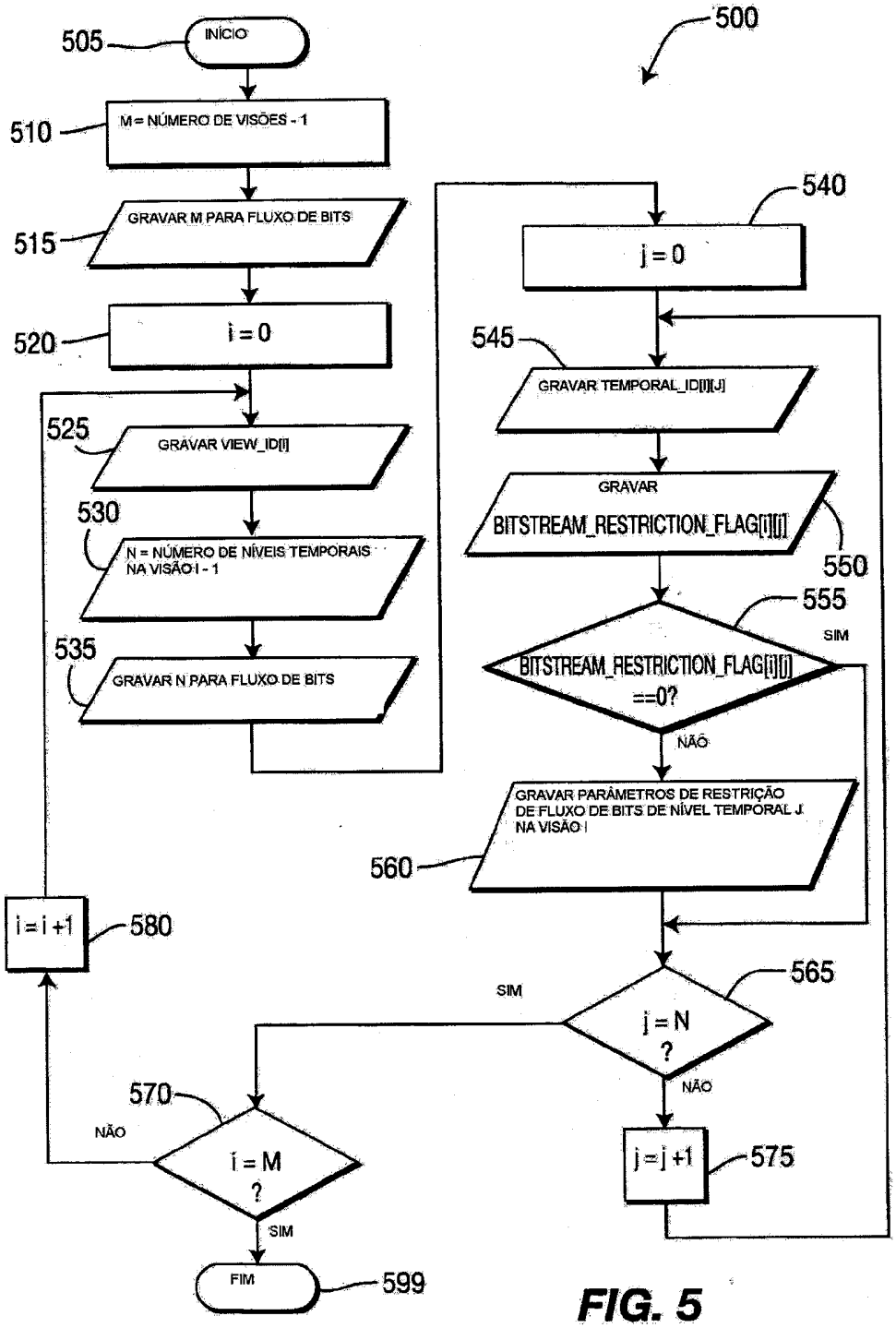

500
505
INÍCIO
510
M = NÚMERO DE VISÕES - 1
515
GRAVAR M PARA FLUXO DE BITS
520
i = 0
525
GRAVAR VIEW_ID[i]
530
N = NÚMERO DE NÍVEIS TEMPORAIS NA VISÃO I - 1
535
GRAVAR N PARA FLUXO DE BITS
540
j = 0
545
GRAVAR TEMPORAL_ID[i][J]
GRAVAR
BITSTREAM_RESTRICTION_FLAG[i][j]
550
555
SIM
BITSTREAM_RESTRICTION_FLAG[i][j]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

FIG. 5

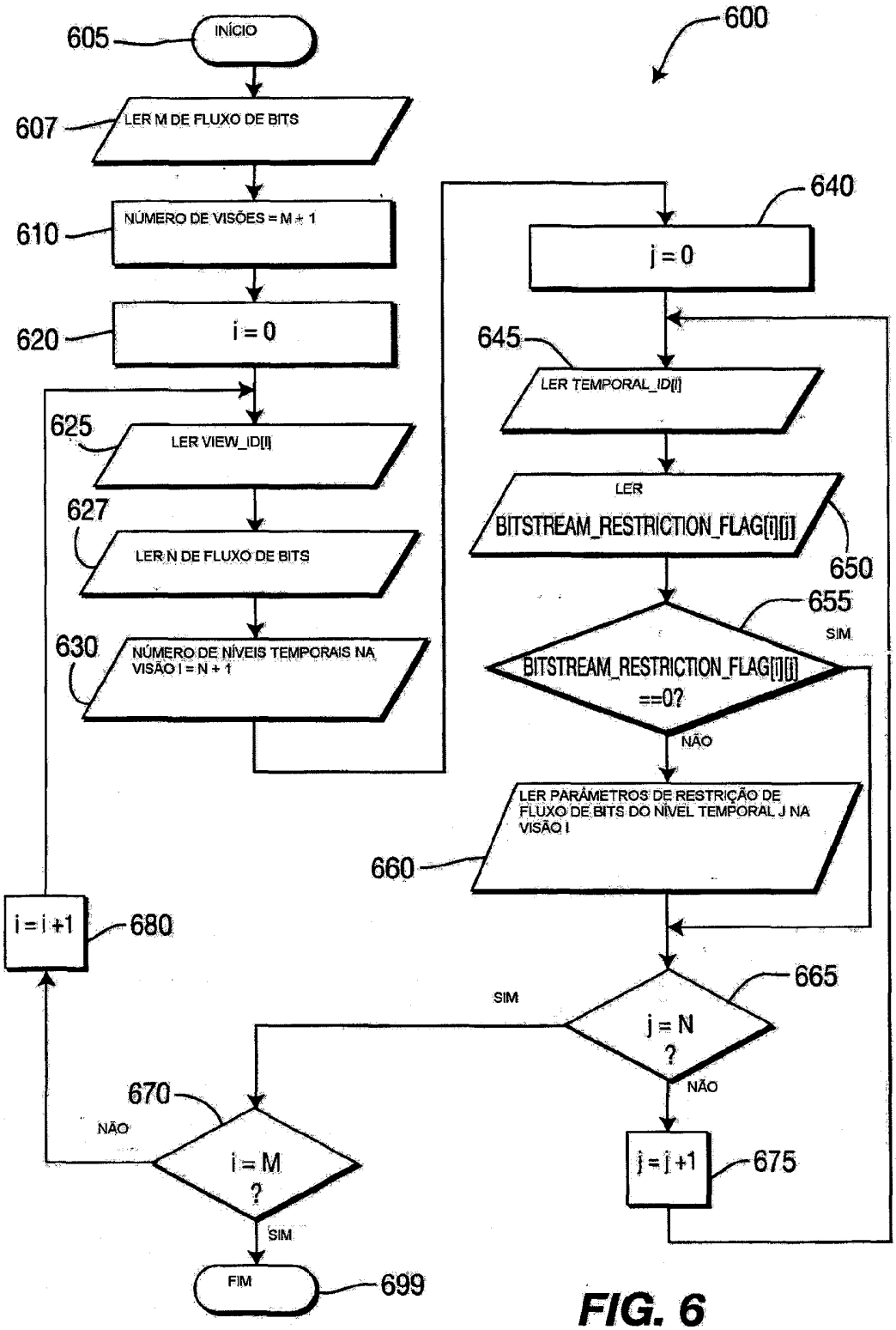

600
605
INÍCIO
607
LER M DE FLUXO DE BITS
610
NÚMERO DE VISÕES = M + 1
620
i = 0
625
LER VIEW_ID[i]
627
LER N DE FLUXO DE BITS
630
NÚMERO DE NÍVEIS TEMPORAIS NA VISÃO I = N + 1
640
j = 0
645
LER TEMPORAL_ID[i]
LER
BITSTREAM_RESTRICTION_FLAG[i][j]
650
655
SIM
BITSTREAM_RESTRICTION_FLAG[i][j] ==0?
NÃO
LER PARÂMETROS DE RESTRIÇÃO DE FLUXO DE BITS DO NÍVEL TEMPORAL J NA VISÃO I
660
665
SIM
j = N ?
NÃO
670
NÃO
i = M ?
SIM
j = j +1
675
i = i +1
680
FIM
699

FIG. 6

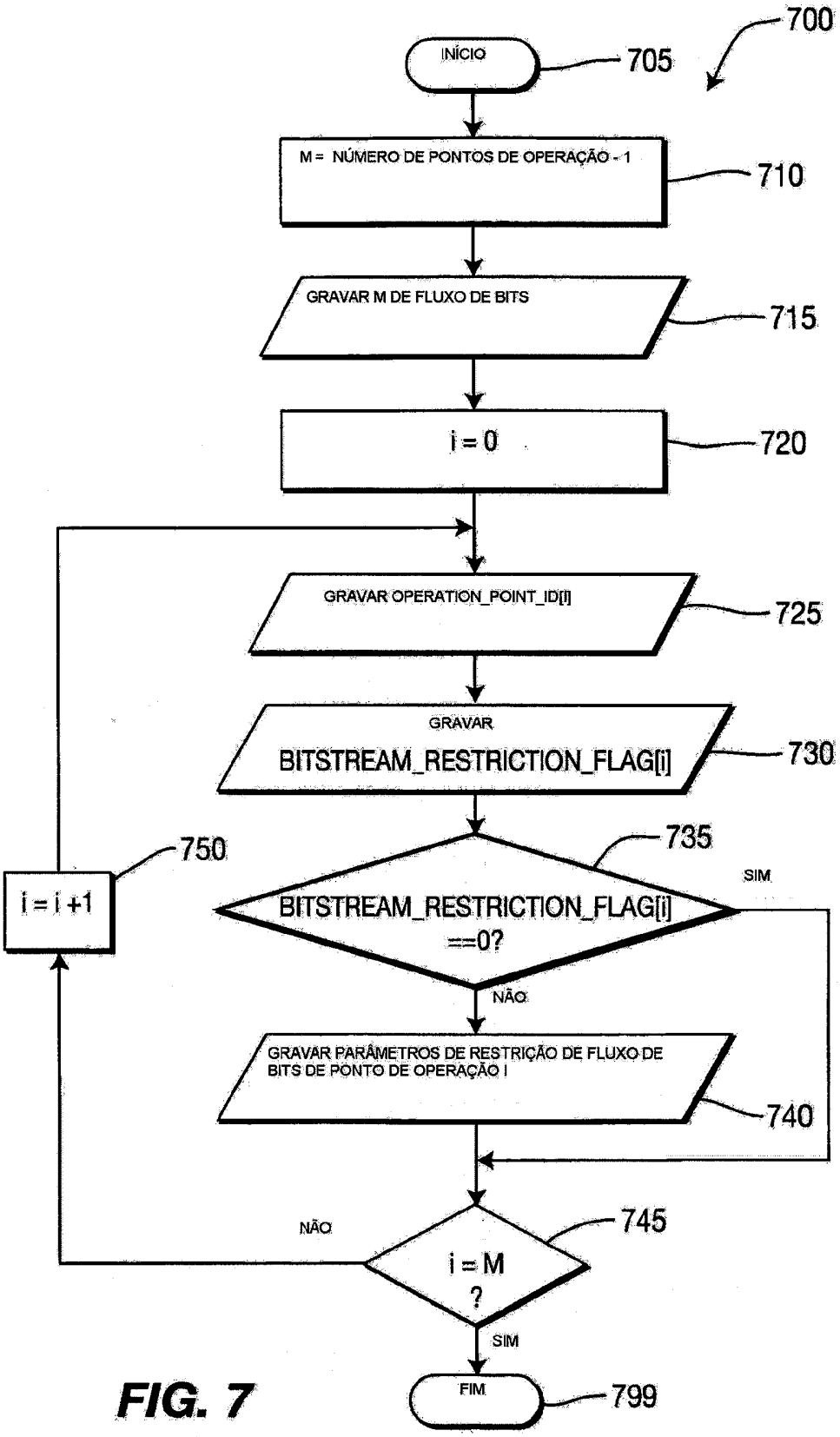

700
INÍCIO
705
M = NÚMERO DE PONTOS DE OPERAÇÃO - 1
710
GRAVAR M DE FLUXO DE BITS
715
i = 0
720
GRAVAR OPERATION_POINT_ID[I]
725
GRAVAR
BITSTREAM_RESTRICTION_FLAG[i]
730
735
BITSTREAM_RESTRICTION_FLAG[i] ==0?
SIM
NÃO
750
i = i +1
GRAVAR PARÂMETROS DE RESTRIÇÃO DE FLUXO DE BITS DE PONTO DE OPERAÇÃO I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

FIG. 7

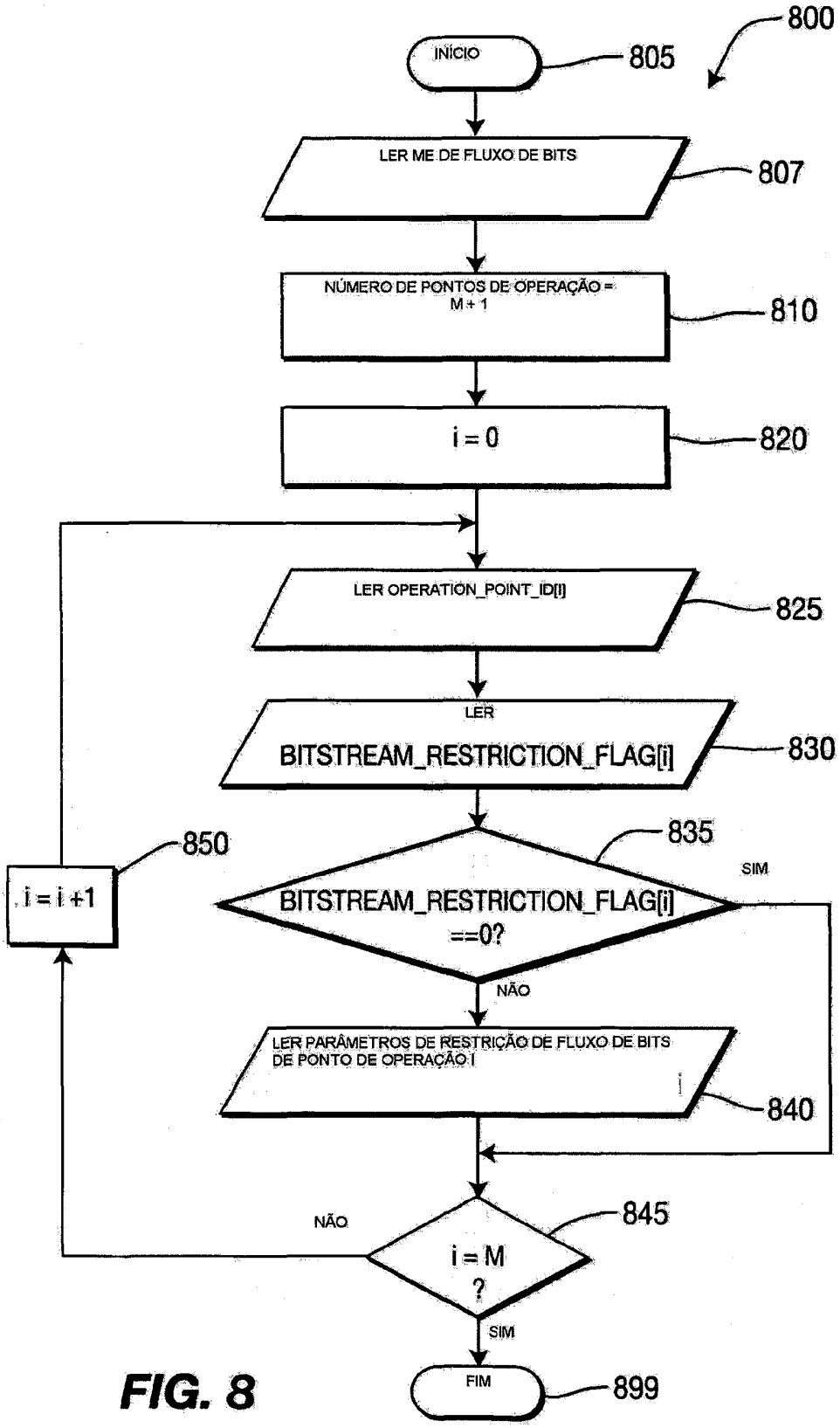

800
INÍCIO
805
LER ME DE FLUXO DE BITS
807
NÚMERO DE PONTOS DE OPERAÇÃO = M + 1
810
i = 0
820
LER OPERATION_POINT_ID[i]
825
LER
BITSTREAM_RESTRICTION_FLAG[i]
830
835
BITSTREAM_RESTRICTION_FLAG[i] ==0?
SIM
NÃO
850
i = i +1
LER PARÂMETROS DE RESTRIÇÃO DE FLUXO DE BITS DE PONTO DE OPERAÇÃO i
i
840
845
NÃO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

FIG. 8

RESUMO

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

São fornecidos métodos e aparelho para incorporação de informação de usabilidade de vídeo (VUI) na Codificação de Vídeo com Múltiplas Visualizações (MVC). Um aparelho (100) inclui um codificador (100) para codificação de conteúdo de vídeo com múltiplas visualizações através da especificação de informação de usabilidade de vídeo para pelo menos um selecionado de: visualizações individuais (300), níveis temporais individuais em uma visualização (500) e pontos operacionais individuais (700). Adicionalmente, um aparelho (200) inclui um decodificador para decodificação de conteúdo de vídeo com múltiplas visualizações através da especificação de informação de usabilidade de vídeo para pelo menos um selecionado de: visualizações individuais (400), níveis temporais individuais em uma visualização (600) e pontos operacionais individuais (800).