

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

(21) PI0609280-2 A2



(22) Data de Depósito: 16/03/2006
(43) Data da Publicação: 09/03/2010
(RPI 2044)

(51) Int.Cl.:
H04N 7/26 (2010.01)
H04N 7/50 (2010.01)
H04N 7/34 (2010.01)

(54) Título: **CODIFICAÇÃO LUMA E CROMA USANDO UM PREDITOR COMUM**

(30) Prioridade Unionista: 13/04/2005 US 60/671.255,
20/07/2005 US 60/700.834

(73) Titular(es): THOMSON LICENSING

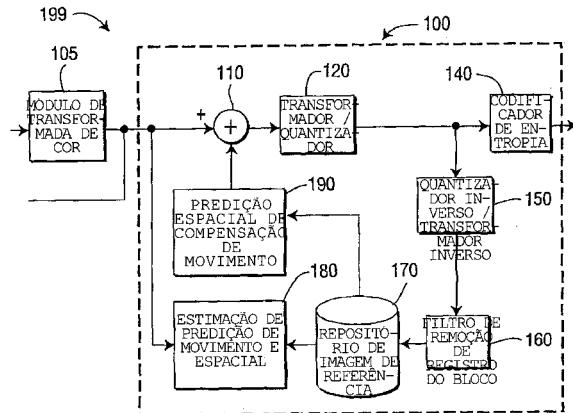
(72) Inventor(es): HAOPING YU

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

(86) Pedido Internacional: PCT US2006009990 de 16/03/2006

(87) Publicação Internacional: WO 2006/113022 de 26/10/2006

(57) Resumo: CODIFICAÇÃO LUMA E CROMA USANDO UM PREDITOR COMUM. São descritos codificadores de vídeo, decodificadores de vídeo e métodos correspondentes. Um codificador de vídeo para codificar dados de sinal de vídeo para um bloco de imagem inclui um codificador (100) para codificar todos os componentes de cor dos dados de sinal de vídeo usando um preditor comum (315). Um decodificador de vídeo para decodificar dados de sinal de vídeo para um bloco de imagem inclui um decodificador (200) para decodificar todos os componentes de cor dos dados de sinal de vídeo usando um preditor comum (430). Adicionalmente, um aparelho e método para codificar e decodificar dados de sinal para um bloco de imagem incluem um codificador e decodificador para codificar / decodificar componentes de cor dos dados de sinal de vídeo sem aplicar uma transformada de cor residual neles. Além do mais, um codificador e decodificador de vídeo para codificar / decodificar dados de sinal de vídeo para um bloco de imagem incluem um codificador e decodificador para codificar / decodificar os dados de sinal de vídeo usando preditores exclusivos para cada um dos componentes de cor dos dados de sinal de vídeo.



"CODIFICAÇÃO LUMA E CROMA USANDO UM PREDITOR COMUM"

REFERÊNCIA CRUZADA A PEDIDOS RELACIONADOS

Este pedido reivindica o benefício do pedido provisório US 60/671.255, depositado em 13 de abril de 2005, e do pedido provisório US 60/700.834, depositado em 20 de julho de 2005, ambos os quais são aqui incorporados pela referência em suas respectivas íntegras. Além do mais, este pedido está relacionado aos pedidos de patente, documentos judiciais PU060023, intitulado "METHOD AND APPARATUS FOR VIDEO ENCODING"; PU060029, PU060030, intitulados "METHOD AND APPARATUS FOR VIDEO DECODING"; e PU050159, intitulado "METHOD AND APPARATUS FOR VIDEO ENCODING AND DECODING", cada um dos quais é depositado concorrentemente com este.

15 CAMPO TÉCNICO

A presente invenção diz respeito, no geral, a codificadores e decodificadores e, mais particularmente, a métodos e aparelhos para codificação e decodificação de vídeo.

ANTECEDENTES DA INVENÇÃO

20 Atualmente, o formato 4:4:4 do padrão H.264 da União Internacional de Telecomunicações, Setor de Telecomunicações (ITU-T) (doravante, o "padrão H.264") somente codifica um de três canais como luma, com os outros dois canais sendo codificados como croma usando ferramentas menos eficientes. Quando uma entrada de um codec está no formato 4:4:4 com resolução completa em todos os componentes de entrada, a codificação de dois dos três componentes de entrada com o algoritmo de codificação croma menos efetivo resulta no uso

de mais bits naqueles dois canais. Este problema particular é mais perceptível em intraquadros. Por exemplo, o padrão H.264 executando no modo somente intra é menos eficiente que JPEG2k para qualidade de compressão completa em 40 dB (PSNR) 5 e superiores.

Dessa maneira, é desejável e altamente vantajoso ter métodos e aparelhos para codificação e decodificação de vídeo que superem as desvantagens supradescritas da tecnologia anterior.

10 SUMÁRIO DA INVENÇÃO

Estes e outros inconvenientes e desvantagens da tecnologia anterior são abordados pela presente invenção, que é direcionada para métodos e aparelhos para codificação e decodificação de vídeo.

15 De acordo com um aspecto da presente invenção, é provido um codificador de vídeo para codificar dados de sinal de vídeo para um bloco de imagens. O codificador de vídeo inclui um codificador para codificar todos os componentes de cor dos dados de sinal de vídeo usando um preditor 20 comum.

De acordo com um outro aspecto da presente invenção, é provido um método para codificar dados de sinal de vídeo para um bloco de imagens. O método inclui codificar todos os componentes de cor dos dados de sinal de vídeo 25 usando um preditor comum.

De acordo com um ainda outro aspecto da presente invenção, é provido um decodificador de vídeo para decodificar dados de sinal de vídeo para um bloco de imagens. O de-

codificador de vídeo inclui um decodificador para decodificar todos os componentes de cor dos dados de sinal de vídeo usando um preditor comum.

De acordo com um ainda outro aspecto da presente invenção, é provido um método para decodificar dados de sinal de vídeo para um bloco de imagens. O método inclui decodificar todos os componentes de cor dos dados de sinal de vídeo usando um preditor comum.

Estes e outros aspectos, recursos e vantagens da presente invenção ficarão aparentes a partir da seguinte descrição detalhada das modalidades exemplares, que deve ser lida em conjunto com os desenhos anexos.

DESCRIÇÃO RESUMIDA DOS DESENHOS

A presente invenção pode ser mais bem entendida de acordo com as seguintes Figuras exemplares, nas quais:

A Figura 1 é um diagrama em blocos ilustrando um aparelho codificador de vídeo exemplar no qual os presentes princípios podem ser aplicados;

A Figura 2 é um diagrama em blocos ilustrando um aparelho decodificador de vídeo exemplar no qual os presentes princípios podem ser aplicados;

A Figura 3 é um fluxograma ilustrando um processo de codificação de vídeo exemplar com um bloco de transformada de cor de pré-codificação de acordo com os presentes princípios;

A Figura 4 é um fluxograma ilustrando um processo exemplar de decodificação de vídeo com um bloco de transformada de cor inversa de pós-decodificação de acordo com os

presentes princípios;

A Figura 5 é um diagrama em blocos ilustrando um modelo simplificado de transformada de cor residual (RCT);

As Figuras 6A e 6B são gráficos de PSNR médio em 5 função da taxa de bit para ATV somente intra de acordo com os presentes princípios;

As Figuras 7A e 7B são gráficos de PSNR médio em função da taxa de bit para CT somente intra de acordo com os presentes princípios;

As Figuras 8A e 8B são gráficos de PSNR médio em função da taxa de bit para DT somente intra de acordo com os presentes princípios;

As Figuras 9A e 9B são gráficos de PSNR médio em função da taxa de bit para MIR_HD somente intra de acordo 15 com os presentes princípios;

As Figuras 10A e 10B são gráficos de PSNR médio em função da taxa de bit para RT somente intra de acordo com os presentes princípios;

As Figuras 11A e 11B são gráficos de PSNR médio em 20 função de taxa de bit para STB_HD somente intra de acordo com os presentes princípios;

A Figura 12 é uma tabela ilustrando a sintaxe do parâmetro da seqüência H.264 de acordo com os presentes princípios;

A Figura 13 é uma tabela ilustrando a sintaxe de 25 dados residuais H.264 de acordo com os presentes princípios;

A Figura 14 é um fluxograma ilustrando um processo de codificação de vídeo exemplar com um bloco de transforma-

da de cor de pré-codificação de acordo com os presentes princípios;

A Figura 15 é um fluxograma ilustrando um processo exemplar de decodificação de vídeo com um bloco de transformada de cor inversa de pós-decodificação de acordo com os presentes princípios; e

A Figura 16 é uma tabela ilustrando a sintaxe de predição de macrobloco H.264 de acordo com os presentes princípios.

10 DESCRIÇÃO DETALHADA

A presente invenção é direcionada para métodos e aparelhos para codificação e decodificação de dados de sinal de vídeo. Percebe-se que, embora a presente invenção seja essencialmente descrita com relação aos dados de sinal de vídeo amostrados usando o formato 4:4:4 do padrão H.264 da União Internacional de Telecomunicações, Setor de Telecomunicações (ITU-T), a presente invenção também pode ser aplicada aos dados de sinal de vídeo amostrados usando outros formatos (por exemplo, o formato 4:2:2 e/ou o formato 4:2:0) do padrão H.264, bem como outros padrões de compressão de vídeo, ao mesmo tempo em que mantém o escopo da presente invenção.

Percebe-se que métodos e aparelhos de acordo com os presentes princípios não exigem o uso de nenhuma ferramenta(s) inédita(s) para o algoritmo de compressão luma ou croma. Em vez disto, podem ser usadas as ferramentas de codificação luma existentes. Dessa maneira, um resultado vantajoso a partir deste ponto é que o desempenho de codifica-

ção do formato 4:4:4 pode ser maximizado ao mesmo tempo em que preserva compatibilidade com versões anteriores e minimiza qualquer mudança do padrão H.264 existente (ou outro aplicável).

5 De acordo com os princípios da presente invenção, da forma configurada em uma modalidade, um algoritmo de codificação luma é usado para codificar todos os três canais componentes, por exemplo, do conteúdo 4:4:4. Vantagens desta modalidade incluem uma melhoria no desempenho geral de codificação para comprimir conteúdo 4:4:4 em relação à tecnologia anterior. Atualmente, no padrão H.264 existente, somente 10 um dos três canais é codificado como luma, e os outros dois são codificados como croma usando ferramentas menos eficientes.

15 Adicionalmente, de acordo com os princípios da presente invenção, da forma configurada em uma modalidade, a transformada de cor é realizada como uma etapa de pré-processamento. Assim, de acordo com esta modalidade, uma transformada de cor residual (RCT) não é realizada dentro do 20 laço de compressão. Vantagens desta modalidade incluem a provisão de consistente arquitetura de codificador / decodificador entre todos os formatos de cor.

Além do mais, de acordo com os princípios da presente invenção, da forma configurada em uma modalidade, o 25 modo de predição de movimento / espacial é usado para todos os três componentes. Vantagens desta modalidade incluem baixa complexidade de codec e compatibilidade com versões anteriores.

Também, de acordo com uma outra modalidade, em vez de usar o mesmo preditor para todos os três componentes, um conjunto (ou subconjunto) de três (3) preditores espaciais restritos pode ser utilizado para os três componentes. Vantagens desta modalidade incluem uma melhoria no desempenho geral de codificação para comprimir conteúdo 4:4:4 em relação à tecnologia anterior.

Percebe-se que várias modalidades supradescritas e, subseqüentemente, aqui descritas, podem ser implementadas como modalidades independentes ou podem ser combinadas de qualquer maneira da forma prontamente percebida pelos versados na técnica. Assim, por exemplo, em uma primeira modalidade combinada, um algoritmo de codificação luma é vantajosamente usado para codificar todos os três canais componentes, a transformada de cor é realizada como uma etapa de pré-processamento, e um único preditor é usado para todos os três canais componentes. Em uma segunda modalidade combinada, um algoritmo de codificação luma é vantajosamente usado para codificar todos os três canais componentes, a transformada de cor é realizada como uma etapa de pré-processamento, e um conjunto (ou subconjunto) de três (3) preditores espaciais restritos pode ser utilizado pelos três canais componentes. Certamente, da forma exposta, outras combinações de várias modalidades também podem ser implementadas dados os preceitos dos presentes princípios aqui providos, embora mantendo o escopo da presente invenção.

A presente descrição ilustra os princípios da presente invenção. Assim, percebe-se que os versados na técnica

serão capazes de desenvolver vários arranjos que, embora não explicitamente aqui descritos ou mostrados, incorporam os princípios da invenção e estão incluídos no seu espírito e escopo.

5 Pretende-se que todos os exemplos e linguagem condicional aqui citados tenham propósitos pedagógicos para auxiliar o leitor no entendimento dos princípios da invenção e dos conceitos contribuídos pelo inventor para promover a tecnologia, e que sejam interpretados sem limitações a tais
10 exemplos e condições especificamente citados.

Além do mais, pretende-se que todas as declarações aqui citando princípios, aspectos e modalidades da invenção, bem como seus exemplos específicos, abranjam tanto seus equivalentes estruturais quanto funcionais. Adicionalmente,
15 pretende-se que tais equivalentes incluam tanto os equivalentes atualmente conhecidos bem como os equivalentes desenvolvidos no futuro, isto é, todos os elementos desenvolvidos que realizem a mesma função, independentemente da estrutura.

Assim, por exemplo, versados na técnica percebem
20 que os diagramas em blocos aqui apresentados representam visões conceituais de conjuntos de circuitos ilustrativos incorporando os princípios da invenção. Similarmente, percebe-se que todos os fluxogramas, diagramas de fluxo, diagramas de transição de estado, pseudocódigos e semelhantes, representam vários processos que podem ser substancialmente representados em meio legível por computador e, portanto, executados por um computador ou processador, esteja este computador ou processador explicitamente mostrado ou não.

As funções dos vários elementos mostrados nas Figuras podem ser providas por meio do uso de hardware dedicado bem como por hardware capaz de executar suporte lógico em associação com suporte lógico apropriado. Quando providas por um processador, as funções podem ser providas por um único processador dedicado, por um único processador compartilhado, ou por uma pluralidade de processadores individuais, alguns dos quais podem ser compartilhados. Além do mais, o uso explícito do termo "processador" ou "controlador" não deve ser interpretado dizendo respeito exclusivamente a hardware capaz de executar suporte lógico e pode incluir implicitamente, sem limitações, hardware processador de sinal digital ("DSP"), memória exclusiva de leitura ("ROM") para armazenar suporte lógico, memória de acesso aleatório ("RAM") e armazenamento não volátil.

Outro hardware, convencional e/ou personalizado, também pode ser incluído. Similarmente, todos os comutadores mostrados nas Figuras são somente conceituais. Suas funções podem ser realizadas 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 da forma mais especificamente entendida pelo contexto.

Nas reivindicações deste, pretende-se que todo elemento expressado como um dispositivo para realizar uma função especificada abrange qualquer maneira de realizar aquela função incluindo, por exemplo, a) uma combinação de elementos de circuito que realiza aquela função ou b) supor-

te lógico em qualquer forma, incluindo, portanto, suporte lógico embarcado, microcódigo ou semelhante, combinado com conjunto de circuito apropriado para executar aquele suporte lógico para realizar a função. A invenção, da forma definida por tais reivindicações, reside no fato de que as funcionalidades providas pelos vários dispositivos citados são combinadas e reunidas na maneira que as reivindicações exigem. Assim, percebe-se que qualquer dispositivo que possa prover aquelas funcionalidades é equivalente àqueles aqui mostrados.

10

Em relação à Figura 1, um aparelho de codificação de vídeo exemplar é indicado, no geral, pelo número de referência 199. O aparelho de codificação de vídeo 199 inclui um codificador de vídeo 100 e um módulo de transformada de cor de pré-codificação 105.

15

O módulo de transformada de cor de pré-codificação 105 serve para realizar um pré-processamento de cor dos sinalis de vídeo antes de dar entrada dos mesmos no codificador de vídeo 100. O pré-processamento de cor realizado pelo módulo de transformada de cor de pré-codificação 105 é adicionalmente aqui descrito a seguir. Percebe-se que o módulo de transformada de cor de pré-codificação 105 pode ser omitido em algumas modalidades.

20

Uma entrada do módulo de transformada de cor de pré-codificação 105 e uma entrada do codificador de vídeo 100 estão disponíveis como entradas do aparelho de codificação de vídeo 199.

Uma saída do módulo de transformada de cor de pré-

codificação 105 é conectada em comunicação de sinal com a entrada do codificador de vídeo 100.

A entrada do codificador de vídeo 100 é conectada em comunicação de sinal com uma entrada não invertida de uma junção de soma 110. A saída da junção de soma 110 é conectada em comunicação de sinal com um transformador / quantizador 120. A saída do transformador / quantizador 120 é conectada em comunicação de sinal com o codificador de entropia 140. Uma saída do codificador de entropia 140 está disponível como uma saída do codificador de vídeo 100 e também como uma saída do aparelho de codificação de vídeo 199.

A saída do transformador / quantizador 120 é adicionalmente conectada em comunicação de sinal com um transformador / quantizador inverso 150. Uma saída do transformador / quantizador inverso 150 é conectada em comunicação de sinal com uma entrada de um filtro de remoção de registro do bloco 160. Uma saída do filtro de remoção de registro do bloco 160 é conectada em comunicação de sinal com os repositórios de imagem de referência 170. Uma primeira saída dos repositórios de imagem de referência 170 é conectada em comunicação de sinal com uma primeira entrada de um estimador de predição de movimento e espacial 180. A entrada do codificador de vídeo 100 é adicionalmente conectada em comunicação de sinal com uma segunda entrada do estimador de predição de movimento e espacial 180. A saída do estimador de predição de movimento e espacial 180 é conectada em comunicação de sinal com uma primeira entrada de um compensador de predição de movimento e espacial 190. Uma segunda saída dos

repositórios de imagem de referência 170 é conectada em comunicação de sinal com uma segunda entrada do compensador de movimento e espacial 190. A saída do compensador de movimento e espacial 190 é conectada em comunicação de sinal com 5 uma entrada invertida da junção de soma 110.

Em relação à Figura 2, um aparelho de decodificação de vídeo é indicado, no geral, pelo número de referência 299. O aparelho de decodificação de vídeo 299 inclui um decodificador de vídeo 200 e um módulo de transformada de cor 10 inversa de pós-decodificação 293.

Uma entrada do decodificador de vídeo 200 está disponível como uma entrada do aparelho de decodificação de vídeo 299. A entrada do decodificador de vídeo 200 é conectada em comunicação de sinal com uma entrada do decodificador de entropia 210. Uma primeira saída do decodificador de entropia 210 é conectada em comunicação de sinal com uma entrada de um quantizador / transformador inverso 220. Uma saída do quantizador / transformador inverso 220 é conectada em comunicação de sinal com uma primeira entrada de uma junção de soma 240. 15 20

A saída da junção de soma 240 é conectada em comunicação de sinal com um filtro de remoção de registro do bloco 290. Uma saída do filtro de remoção de registro do bloco 290 é conectada em comunicação de sinal com repositórios de imagem de referência 250. O repositório de imagem de referência 250 é conectado em comunicação de sinal com uma primeira entrada de um compensador de predição de movimento e espacial 260. Uma saída do compensador de predição de mo- 25

vimento e espacial 260 é conectada em comunicação de sinal com uma segunda entrada da junção de soma 240. Uma segunda saída do decodificador de entropia 210 é conectada em comunicação de sinal com uma segunda entrada do compensador de movimento 260. A saída do filtro de remoção de registro do bloco 290 está disponível como uma saída do decodificador de vídeo 200 e também como uma saída do aparelho de decodificação de vídeo 299.

Além do mais, uma saída do módulo de transformada de cor inversa de pós-decodificação 293 pode estar disponível como uma saída do aparelho de decodificação de vídeo 299. Em um caso como este, a saída do decodificador de vídeo 200 pode ser conectada em comunicação de sinal com uma entrada do módulo de transformada de cor inversa de pós-decodificação 293, que é um módulo de pós-processamento em relação ao decodificador de vídeo 200. Uma saída do módulo de transformada de cor inversa de pós-decodificação 293 provê um sinal transformado de cor inversa pós-processado em relação à saída do decodificador de vídeo 200. Percebe-se que o uso do módulo de transformada de cor inversa de pós-decodificação 293 é opcional.

É agora apresentada uma descrição para melhor codificação 4:4:4 de acordo com os princípios da presente invenção. Uma primeira modalidade descrita é uma modalidade combinada na qual o algoritmo de codificação luma é usado para todos os componentes de cor, o mesmo modo de predição espacial é usado para todos os componentes de cor, e a transformada de cor residual (RCT) é omitida de dentro do

laço de compressão. Resultados de testes para esta modalidade combinada também são providos. Posteriormente, é descrita uma segunda modalidade combinada em que o algoritmo de codificação luma é usado para todos os componentes de cor, um conjunto (ou subconjunto) de preditores espaciais restritos é usado para todos os componentes de cor (em vez de um único modo de predição espacial), e a transformada de cor residual (RCT) é omitida de dentro do laço de compressão. Assim, uma diferença entre a primeira e a segunda modalidades combinadas é o uso de um único modo de predição espacial para todos os componentes de cor na primeira modalidade combinada em oposição ao uso de um conjunto (ou subconjunto) de preditores espaciais restritos para todos os componentes de cor na segunda modalidade combinada. Certamente, diante do exposto, as modalidades aqui descritas podem ser implementadas como modalidades independentes ou podem ser combinadas de qualquer maneira, da forma prontamente percebida pelos versados na técnica. Por exemplo, de acordo com os princípios da presente invenção da forma configurada em ma modalidade, somente um único modo de predição espacial é usado sem combinação com outras modalidades, tal como a omissão da RCT do laço de compressão. Percebe-se que, dados os preceitos dos presentes princípios aqui providos, estas e outras variações, implementações e combinações das modalidades da presente invenção ficarão prontamente determináveis pelos versados na técnica, embora mantendo o escopo da presente invenção.

Em relação à Figura 3, um processo exemplar de codificação de vídeo com um bloco de transformada de cor de

pré-codificação são indicados, no geral, pelos números de referência 300 e 301, respectivamente.

Percebe-se que o bloco de transformada de cor de pré-codificação 301 inclui blocos 306, 308 e 310. Além do mais, percebe-se que o bloco de transformada de cor de pré-codificação 301 é opcional e, assim, pode ser omitido em algumas modalidades da presente invenção.

O bloco de transformada de cor de pré-codificação 301 inclui um bloco de limite de laço 306 que inicia um laço para cada bloco em uma imagem e passa o controle para um bloco de função 308. O bloco de função 308 realiza pré-processamento de cor dos dados de sinal de vídeo do bloco de imagem atual e passa o controle para um bloco de limite de laço 310. O bloco de limite de laço 310 termina o laço. Além do mais, o bloco de limite de laço 310 passa o controle para um bloco de limite de laço 312, este último sendo incluído no processo de codificação de vídeo 300.

O bloco de limite de laço 312 inicia um laço para cada bloco na imagem e passa o controle para um bloco de função 315. O bloco de função 315 forma uma predição de movimento compensado ou espacial do atual bloco de imagem usando um preditor comum para cada componente do atual bloco de imagem e passa o controle para o bloco de função 320. O bloco de função 320 subtrai a predição de movimento compensado ou espacial do atual bloco de imagem para formar uma predição residual e passa o controle para um bloco de função 330. O bloco de função 330 transforma e quantiza a predição residual e passa o controle para um bloco de função 335. O

bloco de função 335 realiza a transformada inversa e quantiza a predição residual para formar uma predição residual codificada e passa o controle para um bloco de função 345. O bloco de função 345 adiciona o residual codificado na predição para formar um bloco de Figura codificado e passa o controle para um bloco de laço final 350. O bloco de laço final 350 termina o laço e passa o controle para um bloco final 355.

Em relação à Figura 4, um processo exemplar de decodificação de vídeo com um bloco de transformada de cor inversa de pós-decodificação são indicados, no geral, pelos números de referência 400 e 460, respectivamente.

Percebe-se que o bloco de transformada de cor inversa de pós-decodificação 460 inclui blocos 462, 464, 466 e 468. Além do mais, percebe-se que o bloco de transformada de cor inversa de pós-decodificação 460 é opcional e, assim, pode ser omitido em algumas modalidades da presente invenção.

O processo de decodificação 400 inclui um bloco de limite de laço 410 que inicia o laço para um bloco atual de uma imagem e passa o controle para um bloco de função 415. O bloco de função 415 faz decodificação de entropia do residual codificado e passa o controle para um bloco de função 420. O bloco de função 420 realiza a transformada inversa e quantiza o residual decodificado para formar um residual codificado e passa o controle para um bloco de função 430. O bloco de função 430 adiciona o residual codificado na predição formada a partir de um preditor comum para cada compo-

nente de cor para formar um bloco de Figura codificado e passa o controle para um bloco de limite de laço 435. O bloco de limite de laço 435 termina o laço e passa o controle para um bloco final 440.

5 Em algumas modalidades, o bloco de limite de laço 435 opcionalmente passa o controle para o bloco de transformada de cor inversa de pós-decodificação 460, em particular, o bloco de limite de laço 462 incluído no bloco de transformada de cor inversa de pós-decodificação 460. O bloco de limite de laço 462 inicia um laço para cada bloco em uma imagem e passa o controle para um bloco de função 464. O bloco de função 464 realiza um pós-processamento de cor inverso dos dados de sinal de vídeo do atual bloco de imagem e passa o controle para um bloco de limite de laço 466. O bloco de limite de laço 466 termina o laço e passa o controle para um bloco final 468.

 No formato 4:4:4 do H.264, cada canal de componente tem resolução completa. Assim, de acordo com a primeira modalidade combinada apresentada anteriormente, o algoritmo 20 de codificação luma é usado em cada componente de cor para alcançar a eficiência geral de compressão ideal. Dessa maneira, na modalidade, para intraquadros, cada componente de cor pode ser comprimido, por exemplo, usando aqueles modos de predição listados na Tabela 8-2, Tabela 8-3 e Tabela 8-4 25 em ISO/IEC 14496 10 Advanced Video Coding 3rd Edition (ITU-T Rec. H.264), ISO/IEC JTC1/SC29/WG1 1 E ITU-T SG16 Q.6, Documento N6540, julho de 2004.

 Além do mais, na modalidade, o mesmo modo de pre-

dição espacial é usado para todos os três componentes de pixel para reduzir adicionalmente a complexidade do codec e melhorar o desempenho. Por exemplo, o modo de predição ajustado pelos parâmetros `prev_intra4x4_pred_mode_flag`, `rem_intra4x4_pred_mode`, `prev_intra8x8_pred_mode_flag`, e `rem_intra8x8_pred_mode` para o luma no cabeçalho de predição do macrobloco pode ser usado por todos os três componentes. Portanto, nenhum bit extra e elementos de sintaxe são necessários. Para os quadros (preditivos) B e P, os pixels de referência em locais fracionários de pixel podem ser calculados para todos os três canais. As mudanças detalhadas de sintaxe e semântica ao padrão H.264 atual são adicionalmente aqui discutidas a seguir.

A transformada de cor residual (RCT) foi adicionada ao codificador / decodificador no Alto Perfil 4:4:4. Em decorrência disto, a estrutura de compressão para o formato 4:4:4 é diferente daquela atualmente usada em todos os outros perfis no padrão H.264 para formatos 4:2:0 e 4:2:2. Isto resulta em alguma complexidade extra na implementação. Além do mais, similar a qualquer outra transformada de cor, YCOCG nem sempre melhora o desempenho geral de compressão. A efetividade do YCOCG é altamente dependente do conteúdo. Assim, para melhorar a compressão e robustez geral, na modalidade, a transformada de cor é colocada fora do laço de predição como uma parte do bloco de pré-processamento. Fazendo isto, selecionar uma transformada de cor ideal para uma tarefa de compressão específica é um problema operacional e a melhor resposta para uma seqüência de entrada particular po-

de ser encontrada entre inúmeras opções. De acordo com uma modalidade em que todos os três componentes estão usando os mesmos preditores espaciais para os intraquadros e os mesmos filtros de interpolação para os quadros B e P (preditivo e 5 intercodificado), ter a transformada de cor realizada nos resíduos de predição é idêntico a realizar a transformada de cor nas imagens fonte fora do codec quando os erros de arredondamento / truncagem são ignorados. Isto será aqui discutido adicionalmente a seguir. Assim, o bloco RCT é removido 10 da estrutura de codificação para tornar a estrutura de codificação consistente entre todos os formatos de cor.

Em relação à Figura 5, um modelo simplificado de RCT é indicado, no geral, pelo número de referência 500. O modelo RCT 500 inclui um gerador de pixel de referência 510, 15 uma junção de soma 520 e um módulo de transformação linear 530. Entradas no gerador de pixel de referência 510 são configuradas para receber informação de movimento / borda e vetores $[X_1]$, $[X_2]$ $[X_n]$. Uma saída do gerador de pixel de referência 510 é conectada em comunicação de sinal 20 com uma entrada invertida da junção de soma 520 que provê o vetor de predição $[X_p]$ nele. Uma entrada não invertida da junção de soma 520 é configurada para receber o vetor de entrada $[X_p]$. Uma saída da junção de soma 520 é configurada em comunicação de sinal com uma entrada do módulo de transformação linear 530, que provê o vetor $[X_d]$ nele. Uma saída do módulo de transformação linear 530 é configurada para prover 25 o vetor $[Y_d]$.

No modelo simplificado da RCT 500, a transformada

de cor representada por uma matriz 3x3 $[A]$ (uma transformação linear) é definida como segue:

$$\begin{bmatrix} Y \\ u \\ v \end{bmatrix} = [A] \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

$[X_n]$, $[X_d]$, $[X_p]$, $[X_f]$, $[X_2]$... $[X_n]$ são vetores 3x1 representando os pixels no domínio RGB. $[Y_d]$ é um vetor 3x1 representando o resultado da transformada de cor. Portanto,

$$[Y_d] = [A][X_d] = [A][X_{in}] - [A][X_p] \quad (2)$$

Posteriormente, na modalidade, os mesmos preditores espaciais e filtros de interpolação são usados para todos os três componentes em um macrobloco de acordo com os princípios da presente invenção da forma configurada em uma modalidade, o pixel de referência $[X_p]$ podendo ser expresso como segue:

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} R_1 & R_2 & R_3 & \dots & R_n \\ G_1 & G_2 & G_3 & \dots & G_n \\ B_1 & B_2 & B_3 & \dots & B_n \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix}, \quad (3)$$

onde um vetor $n \times 1$ $[C]$ representa as operações li-

neares envolvidas nos preditores espaciais e filtros de interpolação definidos no padrão H.264. Aqui, presume-se que o pixel de referência seja calculado usando um número total de n pixels vizinhos $[X_1], [X_2], \dots [X_n]$

5 Substituir $[X_p]$ da equação (3) na equação (2) resulta no seguinte:

$$[Y_d] = [A][X_{in}] - [A] \begin{pmatrix} R_1 & R_2 & R_3 & \dots & R_n \\ G_1 & G_2 & G_3 & \dots & G_n \\ B_1 & B_2 & B_3 & \dots & B_n \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{pmatrix}. \quad (4)$$

Ignorar os erros de arredondamento / truncagem e assumir que o mesmo modo de predição seja selecionado tanto no domínio RGB quanto no domínio Y resulta no seguinte:

$$[Y_d] = [A][X_{in}] - \left([A] \begin{pmatrix} R_1 & R_2 & R_3 & \dots & R_n \\ G_1 & G_2 & G_3 & \dots & G_n \\ B_1 & B_2 & B_3 & \dots & B_n \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{pmatrix} \right) = [Y_{in}] - \begin{pmatrix} Y_1 & Y_2 & Y_3 & \dots & Y_n \\ u_1 & u_2 & u_3 & \dots & u_n \\ v_1 & v_2 & v_3 & \dots & v_n \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{pmatrix}. \quad (5)$$

$$[Y_d] = [Y_{in}] - [Y_1 Y_2 Y_3 \dots Y_n] \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix} \quad (6)$$

Assim, a equação (6) mostra claramente que usar YUV como entrada para o codificador / decodificador, de acordo com os princípios da presente invenção da forma configurada nesta modalidade, é idêntico a realizar RCT.

5 Também, de acordo com os princípios da presente invenção da forma configurada em uma modalidade, um inédito perfil 4:4:4 é adicionado no padrão H.264, aqui referido como "Advanced 4:4:4 Profile with profile_idc=166". Este inédito profile_idc pode ser adicionado no cabeçalho de parâmetro de seqüência e pode ser usado no cabeçalho de camada de 10 macrobloco, bem como no cabeçalho de dados residuais.

Para suportar o uso de algoritmo luma para codificar todos os três componentes de cor, algumas mudanças podem ser feitas na sintaxe de dados residuais. Além do mais, também podem ser feitas mudanças na semântica de alguns dos elementos no cabeçalho do macrobloco, no cabeçalho de dados residuais, e assim por diante. No geral, a sintaxe existente para luma na especificação H.264 permanecerá inalterada e será usada para codificar um dos três componentes. As mudanças são compatíveis com versões anteriores. As mudanças detalhadas de sintaxe e semântica são aqui descritas a seguir.

Agora será dada uma descrição em relação aos re-

sultados da simulação realizada de acordo com os princípios da presente invenção da forma configurada em várias modalidades.

Em relação às Figuras 6A e 6B, gráficos de PSNR 5 médio em função da taxa de bit para ATV somente intra são indicados, no geral, pelos números de referência 600 e 650, respectivamente.

Em relação às Figuras 7A e 7B, gráficos de PSNR 10 médio em função da taxa de bit para CT somente intra são indicados, no geral, pelos números de referência 700 e 750, respectivamente.

Em relação às Figuras 8A e 8B, gráficos de PSNR 15 médio em função da taxa de bit para DT somente intra são indicados, no geral, pelos números de referência 800 e 850.

Em relação às Figuras 9A e 9B, gráficos de PSNR 20 médio em função da taxa de bit para MIRJHD somente intra são indicados, no geral, pelos números de referência 900 e 950, respectivamente.

Em relação às Figuras 10A e 10B, gráficos de PSNR 25 médio em função da taxa de bit para RT somente intra são indicados, no geral, pelos números de referência 1000 e 1050, respectivamente.

Em relação às Figuras 11A e 11B, gráficos de PSNR 30 médio em função da taxa de bit para STBJHD somente intra são indicados, no geral, pelos números de referência 1100 e 1150.

Em particular, as Figuras 6A, 7A, 8A, 9A, 10 e 11A ilustram resultados de teste para o perfil avançado 4:4:4

5 proposto (indicado e precedido pelo termo "inédito") em função de resultados de aproximação correspondentes a ele. Além do mais, as Figuras 6B, 7B, 8B, 9B, 10B e 11B ilustram resultados de teste para o perfil avançado 4:4:4 proposto (indicado e precedido pelo termo "inédito") em função de JPEK2k.

10 Em todas as Figuras 6A, 6B até 11A, 11B, o PSNR é indicado em decibéis (dB) e a taxa de bit é indicada em bits por segundo (bps). ATV, CT, DT, MIR, RT, STB são os nomes dos clipes de teste.

15 Todas as seqüências de teste JVT/FRExt descritas em JVT-J042, Seqüências de Teste Originadas de Filme, foram usadas nos testes. Todas elas são material de filme 4:4:4 de 10 bits e cada clipe tem 58 quadros.

20 20 Os perfis avançados 4:4:4 propostos foram implementados no suporte lógico de referência JVT JM9.6. Tanto a estrutura de codificação somente intra quanto a estrutura de codificação IBBP foram usadas nos testes. O parâmetro de quantização foi ajustado em 6, 12, 18, 24, 30 e 42 para cada uma das curvas R-D. A seleção do modo otimizado em RD foi usada.

25 O perfil avançado 4:4:4 proposto também foi comparado com os resultados que foram conseguidos executando o suporte lógico de referência com o TUVFormat=0(4:0:0) em cada componente de entrada individual. Três contagens de bit comprimido individual separado foram simplesmente somadas para obter o total de bits comprimidos para calcular a taxa de bit comprimido.

Em relação a JPEG2k, o suporte lógico KaKadu v2.2.3 foi usado nos testes. Os resultados do teste foram gerados usando 5 níveis de descompressão de ondeletas com o filtro de ondeletas biortogonal de 9/7 derivações. Havia só 5 mente um ladrilho por quadro e a Otimização em RD para uma dada taxa alvo também foi usada.

Todas as medidas PSNR foram feitas no domínio RGB.

PSNR médio, definido como $(\text{PSNR}(\text{red}) + \text{PSNR}(\text{green}) + \text{PSNR}(\text{blue})) / 3$, é usado para comparar 10 a qualidade geral de compressão. Isto ocorre, principalmente, em virtude de os dados comprimidos em JPEG2k serem computados usando um algoritmo de controle de taxa desconhecido provido pelo suporte lógico. Para alguns casos, os valores 15 RGB PSNR são completamente distantes um do outro, especialmente quando a transformada de cor JPEG2k foi usada.

A comparação de compressão foi realizada como segue:

* New1: o perfil avançado 4:4:4 proposto com um único modo de predição.

20 * New3: o perfil avançado 4:4:4 proposto com três modos de predição.

* RCT-OFF: entrada RGB com RCT=off.

* RCT-ON: entrada RGB com RCT=on.

25 * YCOCG: conversão RGB para YCOCG foi feita fora do codec. Então, o YCOCG convertido foi usado como a entrada para o suporte lógico JVT.

* R+G+B: Método proposto aproximado comprimindo os sinais R, G e B separadamente.

* Y+CO+CG: Método proposto aproximado comprimindo os sinais convertidos Y, CO, CG separadamente.

*J2k_RGB: A compressão JPEG2k foi feita no domínio RGB. A transformada de cor JPEG2k foi desligada.

5 *J2k_YUV: A compressão JPEG2k foi feita no domínio YUV. A transformada de cor JPEG2k foi usada.

De acordo com os resultados do teste, uma implementação de acordo com os princípios da presente invenção da forma configurada em uma modalidade, no geral, é muito similar ao JPEG2k em termos de eficiência geral de compressão. 10 Em alguns casos, ela é ainda ligeiramente melhor.

Adicionalmente, uma implementação de acordo com os princípios da presente invenção da forma configurada em uma modalidade provê desempenho (compressão) significativamente melhor que o atual Alto Perfil 4:4:4 para qualidade acima de 15 40 dB (PSNR). Especificamente, New1-YCOCG ou New3-YCOCG é melhor que YCOCG e RCT-ON; New1-RGB ou New3-RGB é melhor que RCT-OFF. Em um PSNR igual ou maior que 45 dB (PSNR), a melhoria média no PSNR médio é maior que 1,5 dB. No último exemplo, a melhoria pode ser traduzida em mais de 25% de economia de bit em um PSNR igual a 45 dB. 20

De acordo com os resultados do teste, parece que as transformações de cor ajudam no desempenho de codificação quando o conteúdo é mais saturado de cor, tais como TP, RT. 25 Isto é, se a cor for neutra e menos saturada, codificar o domínio RGB pode ser a escolha certa. A observação exposta é independente de qual transformada de cor é usada.

Comparando os resultados de New1-YCOCG ou New3-

5 YCOCG e JPEG-2k_YUV, observou-se que o desempenho de uma transformada de cor específica em termos de melhorar a eficiência de codificação é muito dependente do conteúdo. Nem sempre a transformada de cor única é melhor. Portanto, nos-
10 sos dados confirmaram que ter uma transformada de cor, tal como RCT, dentro de um laço de codificação (ou decodificação) pode não ser uma boa idéia. Em vez disto, realizar a transformada de cor, se for necessária, fora do codificador / decodificador pode fazer com que todo o sistema de compressão proveja um desempenho melhor e mais robusto.

15 Comparando YCOCG com RCT-ON, os resultados do teste não mostram nenhuma melhoria na eficiência de codificação em relação a RCT. Além do mais, percebe-se que executar o suporte lógico de referência com a RCT ligada aumentou significativamente o tempo de codificação. O tempo de execução foi mais de 2,5 vezes maior.

20 Agora, será dada uma descrição em relação a mudanças de sintaxe e semântica de acordo com os princípios da presente invenção da forma configurada em uma modalidade.

25 Em relação à Figura 12, uma tabela para sintaxe de parâmetro de seqüência H.264 é indicada, no geral, pelo número de referência 1200. Mudanças na sintaxe de acordo com os princípios da presente invenção da forma configurada em uma modalidade são indicadas por texto em itálico.

Em relação à Figura 13, uma tabela para sintaxe de dados residuais H.264 é indicada, no geral, pelo número de referência 1300. Adições / mudanças na sintaxe de acordo com os princípios da presente invenção da forma configurada em

uma modalidade são indicadas por texto em itálico. Na tabela 1300, a seção luma no cabeçalho de dados residuais junto com algumas modificações de texto necessárias são repetidas duas vezes para suportar o lumal e luma2, respectivamente.

Conforme exposto, a supradescrita primeira modalidade combinada foi avaliada e testada implementando os presentes princípios no suporte lógico de referência JVT JM9.6. Os resultados de teste marcados com New1-RGB ou New1-YCOCG representam a primeira modalidade combinada.

Conforme exposto, de acordo com os princípios da presente invenção da forma configurada em uma modalidade, um conjunto (ou subconjunto) de três (3) preditores espaciais restritos é utilizado para os canais componentes (por exemplo, formatos RGB, YUV, YCrCb, e assim por diante) em vez de um único modo de predição espacial. Além do mais, conforme exposto, esta modalidade pode ser combinada com outras modalidades aqui descritas, tais como, por exemplo, o uso de somente o algoritmo de codificação luma para codificar todos os três canais componentes de conteúdo e/ou o uso de transformada de cor como uma etapa de pré-processamento.

Agora, será dada uma descrição em relação à supra-descrita segunda modalidade combinada envolvendo o uso de um conjunto (ou subconjunto) de três (3) preditores espaciais restritos para os componentes de cor, o uso de somente o algoritmo de codificação luma para codificar todos os três componentes de cor e o uso de transformada de cor como uma etapa de pré-processamento (isto é, sem RCT no laço de compressão). Algumas variações desta modalidade também serão

descritas com isto.

Em relação à Figura 14, um processo exemplar de codificação de vídeo com uma etapa de transformada de cor de pré-codificação são indicados, no geral, pelos números de referência 1400 e 1401, respectivamente.

Percebe-se que o bloco de transformada de cor de pré-codificação 1401 inclui blocos 1406, 1408 e 1410. Além do mais, percebe-se que o bloco de transformada de cor de pré-codificação 1401 é opcional e, assim, pode ser omitido em algumas modalidades da presente invenção.

O bloco de transformada de cor de pré-codificação 1401 inclui um bloco de limite de laço 1406 que inicia um laço para cada bloco em uma imagem e passa o controle para um bloco de função 1408. O bloco de função 1408 desempenha o pré-processamento de cor dos dados de sinal de vídeo do atual bloco de imagem e passa o controle para um bloco de limite de laço 1410. O bloco de limite de laço 1410 termina o laço. Além do mais, o bloco de limite de laço 1410 passa o controle para um bloco de limite de laço 1412, este último sendo incluído no processo de codificação de vídeo 1400.

O bloco de limite de laço 1412 inicia um laço para cada bloco na imagem e passa o controle para um bloco de função 1415. O bloco de função 1415 forma uma predição de movimento compensada ou espacial do atual bloco de imagem usando um preditor comum para cada componente de cor do atual bloco de imagem e passa o controle para um bloco de função 1420. O bloco de função 1420 subtrai a predição de movimento compensada ou espacial da atual bloco de imagem para

formar uma predição residual e passa o controle para um bloco de função 1430. O bloco de função 1430 transforma e quantiza a predição residual e passa o controle para um bloco de função 1435. O bloco de função 1435 realiza a transformada inversa e quantiza a predição residual para formar uma predição residual codificada e passa o controle para um bloco de função 1445. O bloco de função 1445 adiciona o residual codificado na predição para formar um bloco de Figura e passa o controle para um bloco de laço final 1450. O bloco de laço final 1450 termina o laço e passa o controle para um bloco final 1455.

Em relação à Figura 15, um processo exemplar de decodificação de vídeo com uma etapa de transformada de cor inversa de pós-decodificação são indicados, no geral, pelos 15 números de referência 1500 e 1560, respectivamente.

Percebe-se que o bloco de transformada de cor inversa de pós-decodificação 1560 inclui blocos 1562, 1564, 1566 e 1568. Além do mais, percebe-se que o bloco de transformada de cor inversa de pós-decodificação 1560 é opcional 20 e, assim, pode ser omitido em algumas modalidades da presente invenção.

O processo de decodificação 1500 inclui um bloco de limite de laço 1510 que inicia o laço para um bloco atual de uma imagem e passa o controle para um bloco de função 25 1515. O bloco de função 1515 faz decodificação de entropia do residual codificado e passa o controle para um bloco de função 1520. O bloco de função 1520 realiza a transformada inversa e quantiza o residual decodificado para formar um

residual codificado e passa o controle para um bloco de função 1530. O bloco de função 1530 adiciona o residual codificado na predição formada a partir de um preditor comum para cada componente de cor para formar um bloco de Figura codificado e passa o controle para um bloco de limite de laço 1535. O bloco de limite de laço 1535 termina o laço e passa o controle para um bloco final 1540.

Em algumas modalidades, o bloco de limite de laço 1535 opcionalmente passa o controle para o bloco de transformada de cor inversa de pós-decodificação 1560, em particular, o bloco de limite de laço 1562 incluído no bloco de transformada de cor inversa de pós-decodificação 1560. O bloco de limite de laço 1562 inicia um laço para cada bloco em uma imagem e passa o controle para um bloco de função 1564. O bloco de função 1564 realiza um pós-processamento de cor inversa dos dados de sinal de vídeo do atual bloco de imagem e passa o controle para um bloco de limite de laço 1566. O bloco de limite de laço 1566 termina o laço e passa o controle para um bloco final 1568.

Conforme exposto, é divulgado um perfil inédito (profile_idc=166) para o perfil avançado 4:4:4. Este perfil inédito também pode ser usado para a segunda modalidade combinada com mudanças de semântica e sintaxe correspondentes da forma aqui descrita a seguir para a segunda modalidade combinada. Este inédito profile_idc é adicionado no Conjunto de Parâmetro de Seqüência e será usado, principalmente, nos cabeçalhos subseqüentes para indicar que o formato de entrada é 4:4:4 e todos os três canais de entrada são codificados

de forma similar a luma.

Para minimizar as mudanças necessárias ao padrão H.264, nenhum tipo inédito de macrobloco é divulgado para o Perfil avançado 4:4:4. Em vez disto, todos os tipos de macrobloco juntos com os parâmetros de codificação associados listados na Tabela 7-11, Tabela 7-13 e Tabela 7-14 do padrão H.264 ainda são válidos. Para o caso de intramacroblocos, todos os três canais de entrada, luma, Cr e Cb serão codificados com base no MbPartPredMode definido na Tabela 7-11 do padrão H.264. Por exemplo, um macrobloco Intra_4x4 no perfil avançado 4:4:4 significa que todos os canais componentes de entrada podem ser codificados usando todos os 9 possíveis modos de predição dados na Tabela 8-2 do padrão H.264. Para referência, na atual Alto Padrão 4:4:4, dois dos canais para um macrobloco Intra_4x4 serão tratados como croma e somente um dos 4 possíveis modos de intrapredição na Tabela 8-5 do padrão H.264 será usado. Para os macroblocos B e P, as mudanças feitas no padrão avançado 4:4:4 ocorrem no processo de interpolação para o cálculo do valor de pixel de referência no local fracionário de pixel. Aqui, o procedimento descrito na Seção 8.4.2.2.1 do padrão H.264, processo de interpolação de amostra luma, será aplicado para luma, Cr e Cb. Novamente para referência, o atual Alto Padrão 4:4:4 usa a Seção 8.4.2.2.2 do padrão H.264, processo de interpolação de amostra croma, para dois dos canais de entrada.

No caso em que o CABAC é escolhido como o modo de codificação de entropia, dois conjuntos separados de modelos de contexto idênticos àqueles atualmente definidos para luma

serão criados para Cr e Cb. Eles também serão atualizados independentemente durante o curso da codificação.

Finalmente, na modalidade, uma vez que não há bloco RCT no laço de codificação, a ResidueColorTransformFlag é 5 removida do conjunto de parâmetro de seqüência no perfil avançado 4:4:4.

Até este ponto, a maioria das mudanças de sintaxe ocorre nos dados residuais da forma mostrada na Figura 13, onde a sintaxe original para luma é repetida duas vezes para 10 suportar Cr e Cb nos perfis avançados 4:4:4 propostos.

Em relação à tabela de camada de macrobloco H.264 (não mostrada), mudanças de semântica na sintaxe correspondente incluem o seguinte.

coded_block_pattern (Add). Quando croma_format_idc 15 for igual a 3 e coded_block_pattern estiver presente, CodedBlockPatternCroma deve ser ajustado em 0. Além do mais, CodedBlockPatternLuma especifica, para cada um dos doze blocos 8x8 luma, Cb e Cr do macrobloco, um dos seguintes casos: (1) Todos os níveis de coeficiente da transformada dos doze blocos 20 luma 4x4 nos blocos luma 8x8, Cb 8x8 e Cr 8x8 são iguais a zero; (2) Um ou mais níveis de coeficiente da transformada de um ou mais dos blocos luma 4x4 nos blocos luma 8x8, Cb 8x8 e Cr 8x8 devem receber valor não zero.

Agora, será dada uma descrição em relação à seleção 25 do modo de predição espacial para os intrablocos de acordo com a segunda modalidade combinada (ou a única modalidade relacionada ao uso do conjunto (ou subconjunto) de três preditores espaciais restritos).

Para cada componente escolher seu melhor MbPartPredMode e o subseqüente melhor modo de predição espacial independentemente, como no caso enquanto está codificando cada canal de entrada separadamente, alguns tipos inéditos de intrablocos podem ser adicionados na Tabela 7-11 do padrão H.264. Em decorrência disto, será feita uma grande quantidade de mudanças no padrão H.264. Em uma modalidade relacionada à segunda modalidade combinada, o mbjypes atual permanece inalterado e é provida uma solução alternativa. Na modalidade, os três canais de entrada são restritos para ser codificados com o mesmo MbPartPredMode ou tipo de macrobloco. Então, uma pequena quantidade de elementos inéditos é adicionada na Sintaxe de Predição do Macrobloco para suportar três modos de predição separados. Portanto, cada componente ainda pode, teoricamente, escolher seu melhor modo de predição espacial independentemente a fim de minimizar o erro de predição para cada canal componente. Por exemplo, assumindo que um macrobloco Intra_4x4 seja escolhido como o mbjype, luma, Cr ou Cb ainda podem encontrar seus próprios melhores modos de predição espacial na Tabela 8-2 na Seção 8.3.1.1 do padrão H.264, tal como, por exemplo, Intra_4x4_Vertical para luma, Intra_4_4_Horizontal para Cr e Intra_4x4_Diagonal_Down_Left para Cb.

Uma outra abordagem, em relação à primeira modalidade combinada supradescrita, é compelir todos os três canais de entrada a compartilhar o mesmo modo de predição. Isto pode ser feito usando a informação de predição que é atualmente suportado pelos elementos de sintaxe existentes,

tais como `prev_intra4x4_pre_mode_flag`, `rem_intra4x4_pred_mode`, `pre_intra8x8_pred_mode_flag` e `rem_intra8x8_pred_mode`, na sintaxe de Predição do Macrobloco. Esta opção irá resultar em menos mudanças no padrão H.264 e, também, em alguma ligeira perda de eficiência de codificação.

Com base nos resultados de testes, usar três modos de predição pode melhorar o desempenho geral de codificação em cerca de 0,2 dB em relação à primeira modalidade combinada.

Em relação à Figura 16, uma tabela para a sintaxe de predição do macrobloco H.264 é indicada, no geral, pelo número de referência 1700. Para referência, a Sintaxe de Predição de Macrobloco modificada para suportar o uso de três modos de predição é listada a seguir, onde:

`prev_intra4x4_pred_mode_flag0` e
`rem_intra4x4_pred_mode0` são para luma;
`prev_intra4x4_pred_mode_flag1` e
`rem_intra4x4_pred_mode1` são para Cr;
`prev_intra4x4_pred_mode_flag2` e
`rem_intra4x4_pred_mode2` são para Cb.

Agora, será dada uma descrição em relação aos resultados da simulação realizada de acordo com os princípios da presente invenção da forma configurada em uma modalidade, para a segunda modalidade combinada.

Todas as seqüências de teste JVT/FRExt são descritas em JVT-J042, Seqüências de Teste Originadas de Filme, JVT-J039 (Viper). Todas elas são materiais 4:4:4 de 10 bits

e cada clipe tem 58 quadros.

O algoritmo proposto foi implementado no suporte lógico de referência JVT JM9.6 e o suporte lógico modificado foi usado nos testes. Tanto somente intra quanto IBRrBP foram testados. Aqui, "Br" significa as Figuras B gravadas. O caso somente intra foi feito para todas as seqüências com o parâmetro de quantização igual a 6, 12, 18, 24, 30, 36 e 42. Em função da grande quantidade de tempo envolvido na simulação, a estrutura IBRrBP GOP somente foi feita para os clipes de filme com um parâmetro de quantização igual a 12, 18, 24, 30 e 36. De acordo com a discussão no AHG 4:4:4, os seguintes parâmetros chave foram usados nos testes:

```
SymbolMode = 1
RDOptimization = 1
15 ScalingMatrixPresentFlag = 0
OffsetMatrixPresentFlag = 1
QoffsetMatrixFile = "q_offset.cfg"
AdaptiveRounding = 1
AdaptRnd Period = 1
AdaptRndCroma = 1
20 AdaptRndWFactorX = 8
SearchRange = 64
UseFME = 1
```

Em relação a JPEG2k, o suporte lógico KaKadu

25 V2.2.3 foi usado nos testes. Os resultados do teste foram gerados usando 5 níveis de descompressão de ondeletas com o filtro de ondeletas biortogonal de 9/7 derivações. Havia somente um ladrilho por quadro e a Otimização em RD para uma

dada taxa alvo também foi usada.

As medições PSNR foram calculadas primariamente no domínio de cor original dos conteúdos fonte, que é RGB para os clipes supradescritos. PSNR médio, definido como $(\text{PSNR(red)} + \text{PSNR(green)} + \text{PSNR(blue)})/3$, é usado para comparar a qualidade geral de compressão.

A comparação de compressão foi realizada como segue:

10 New1: o Perfil avançado 4:4:4 proposto com um único modo de predição.

New3: o Perfil avançado 4:4:4 proposto com três modos de predição.

RCT-OFF: entrada RGB com RCT=off.

RCT-ON: entrada RGB com RCT=on.

15 YCOCG: conversão RGB para YCOCG foi feita fora do codec. Então, o YCOCG convertido foi usado como a entrada para o suporte lógico JVT.

R+G+B: Método proposto aproximado comprimindo os sinais R, G e B separadamente.

20 Y+CO+CG: Método proposto aproximado comprimindo os sinais Y, CO e CG convertidos separadamente.

JPEG2K_RGB: A compressão JPEG2k foi feita no domínio RGB. A transformada de cor JPEG2k foi desligada.

25 JPEG2k_YUV: A compressão JPEG2k foi feita no domínio YUV. A transformada de cor JPEG2k foi usada.

Para o caso somente intra, o perfil avançado 4:4:4 proposto de acordo com os presentes princípios é muito similar ao JPEK2k em termos de eficiência geral de compressão.

Em alguns casos, ele é mesmo ligeiramente melhor.

A abordagem de acordo com os princípios da presente invenção é claramente melhor que o atual Alto Perfil 4:4:4. Em um PSNR igual e maior que 45 dB (PSNR), a melhoria média no PSNR médio é maior que 1,5 dB. Em alguns casos, a melhoria pode ser traduzida em mais de 25% de economia de bit em um PSNR igual a 45 dB.

10 Agora, será dada uma descrição de alguns dos muitos recursos / vantagens presentes providos pelos princípios das modalidades da presente invenção.

Os resultados do teste demonstram que o perfil avançado 4:4:4 proposto utilizando as melhorias correspondentes aos princípios da presente invenção proporciona melhor desempenho comparado com o atual Alto Perfil 4:4:4. O ganho de desempenho é significativo. Além do mais, mover a transformada de cor para fora do codec tornará a arquitetura do codec consistente entre todos os formatos de cor. Em decorrência disto, isto torna a implementação mais fácil e reduz o custo. Isto também torna o codec mais robusto em termos de selecionar a transformada de cor ideal para alcançar melhor eficiência de codificação. Também, a abordagem proposta não adiciona nenhuma ferramenta de codificação inédita e exige somente algumas ligeiras mudanças na sintaxe e na semântica.

25 Assim, de acordo com os princípios da presente invenção da forma configurada em uma modalidade, um método e aparelho são providos para codificação e decodificação de vídeo. São providas modificações no padrão H.264 existente que melhoram o desempenho além daquele atualmente alcançá-

vel. Além do mais, o desempenho é melhorado ainda além do JPEG-2000 para aplicações de alta qualidade. De acordo com os princípios da presente invenção da forma configurada em uma modalidade, melhorias significativas de desempenho de 5 codificação 4:4:4 no padrão H.264 podem ser alcançadas usando o algoritmo de codificação luma para codificar todos os três componentes de cor do conteúdo 4:4:4. Isto é, nenhuma ferramenta inédita é necessária para o algoritmo de codificação luma (ou croma, que não é usado). Em vez disto, ferramentas de codificação luma existentes são utilizadas. Adicionalmente, mudanças de sintaxe e de semântica ao atual perfil 4:4:4 podem ser implementadas de acordo com os presentes princípios para suportar a codificação luma de todos os três canais componentes. Em testes conduzidos de acordo 10 com uma modalidade da presente invenção, quando o conteúdo fonte tem muitas texturas e bordas espaciais, as ferramentas de predição espacial usadas em luma, exibiram claramente seu desempenho superior àquelas usadas em croma. Para algumas das seqüências de teste, quando cada componente de cor foi 15 codificada como luma, foi observada mais de 30% de redução de bit em uma qualidade de compressão maior ou igual a 45 dB (PSNR médio).

Percebe-se que, embora a presente invenção tenha sido aqui descrita essencialmente em relação aos dados de 20 sinal de vídeo amostrados usando o formato 4:4:4 do padrão H.264, a presente invenção também pode ser prontamente implementada em relação aos dados de sinal de vídeo amostrados usando outros formatos (por exemplo, o formato 4:2:0 e/ou o

formato 4:2:2) do padrão H.264, bem como outros padrões de compressão de vídeo. Dados os preceitos da presente invenção aqui providos, estas e outras variações da presente invenção também podem ser prontamente implementadas pelos versados na 5 técnica, embora mantendo o escopo da presente invenção.

Estes e outros recursos e vantagens da presente invenção podem ser prontamente determinados pelos versados na técnica com base nos preceitos aqui divulgados. Entende-se que os preceitos da presente invenção podem ser implementados em várias formas de hardware, suporte lógico, suporte 10 lógico embarcado, processadores com propósito especial ou suas combinações.

Mais preferivelmente, os preceitos da presente invenção são implementados como uma combinação de hardware e suporte lógico. Além do mais, o suporte lógico pode ser implementado como um programa de aplicação embutido de forma tangível em uma unidade de armazenamento de programa. O programa de aplicação pode ser carregado em uma máquina compreendendo qualquer arquitetura adequada e executado por ela. 15 Preferivelmente, a máquina é implementada em uma plataforma de computador com hardware tais 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 operacional e código de microinstrução. Os vários processos e funções aqui descritos podem ser tanto parte do código de 20 microinstrução quanto parte do programa de aplicação, ou qualquer combinação destes, que podem ser executadas por uma 25

CPU. Além do mais, várias outras unidades periféricas podem ser conectadas na plataforma do computador, tais como uma unidade de armazenamento de dados adicional e uma unidade de impressão.

5 Entende-se adicionalmente que, em virtude de alguns dos componentes e métodos constituintes do sistema representado nos desenhos anexos ser, preferivelmente, implementados em suporte lógico, as conexões reais entre os componentes do sistema ou os blocos de função do processo podem 10 diferir dependendo da maneira na qual a presente invenção é programada. Dados os preceitos aqui divulgados, versados na técnica serão capazes de contemplar estas e similares implementações ou configurações da presente invenção.

15 Embora as modalidades ilustrativas tenham sido aqui descritas em relação aos desenhos anexos, entende-se que a presente invenção não está limitada a estas precisas modalidades e que várias mudanças e modificações podem ser nelas realizadas pelos versados na técnica sem fugir do escopo e do espírito da presente invenção. Pretende-se que todas tais 20 mudanças e modificações estejam incluídas no escopo da presente invenção da forma apresentada nas reivindicações anexas.

REIVINDICAÇÕES

1. Codificador de vídeo para codificar dados de sinal de vídeo para um bloco de imagem, **CARACTERIZADO** pelo fato de que o codificador de vídeo comprehende um codificador (100) para codificar todos os componentes de cor dos dados de sinal de vídeo usando um preditor comum.

5 2. Codificador de vídeo, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o preditor comum é um preditor luma usado tanto por componentes luma quanto por componentes croma dos dados de sinal de vídeo.

10 3. Codificador de vídeo, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o dito codificador (100) usa um modo de predição espacial comum para todos os componentes de cor dos dados de sinal de vídeo.

15 4. Codificador de vídeo, de acordo com a reivindicação 3, **CARACTERIZADO** pelo fato de que o modo de predição espacial comum é ajustado pelos parâmetros prev_intra_8x8_pred_mode_flag, rem_intra8x8_pred_mode, prev_intra4x4_pred_mode_flag e rem_intra4x4_pred_mode do padrão H.264 da União Internacional de Telecomunicações, Setor de Telecomunicações.

20 5. Codificador de vídeo, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o dito codificador (100) usa filtros de interpolação comuns para quadros B e P para todos os componentes de cor dos dados de sinal de vídeo.

25 6. Codificador de vídeo, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o dito codificador

(100) codifica todos os componentes de cor dos dados de sinal de vídeo sem aplicar uma transformada de cor residual neles.

7. Codificador de vídeo, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que a amostragem dos dados de sinal de vídeo corresponde a qualquer um dos formatos 4:4:4, 4:2:2 e 4:2:0 do padrão H.264 da União Internacional de Telecomunicações, Setor de Telecomunicações.

8. Método para codificar dados de sinal de vídeo para um bloco de imagem, **CARACTERIZADO** pelo fato de que o método compreende codificar (315) todos os componentes de cor dos dados de sinal de vídeo usando um preditor comum.

9. Método, de acordo com a reivindicação 8, **CARACTERIZADO** pelo fato de que o preditor comum é um preditor luma usado tanto para componentes luma quanto para componentes croma dos dados de sinal de vídeo.

10. Método, de acordo com a reivindicação 8, **CARACTERIZADO** pelo fato de que um modo de predição espacial comum é usado para todos os componentes de cor dos dados de sinal de vídeo.

11. Método, de acordo com a reivindicação 10, **CARACTERIZADO** pelo fato de que o modo de predição espacial comum é ajustado pelos parâmetros prev_intra8x8_pred_mode_flag, rem_intra8x8_pred_mode, prev_intra4x4_pred_mode_flag e rem_intra4x4_pred_mode do padrão H.264 da União Internacional de Telecomunicações, Setor de Telecomunicações.

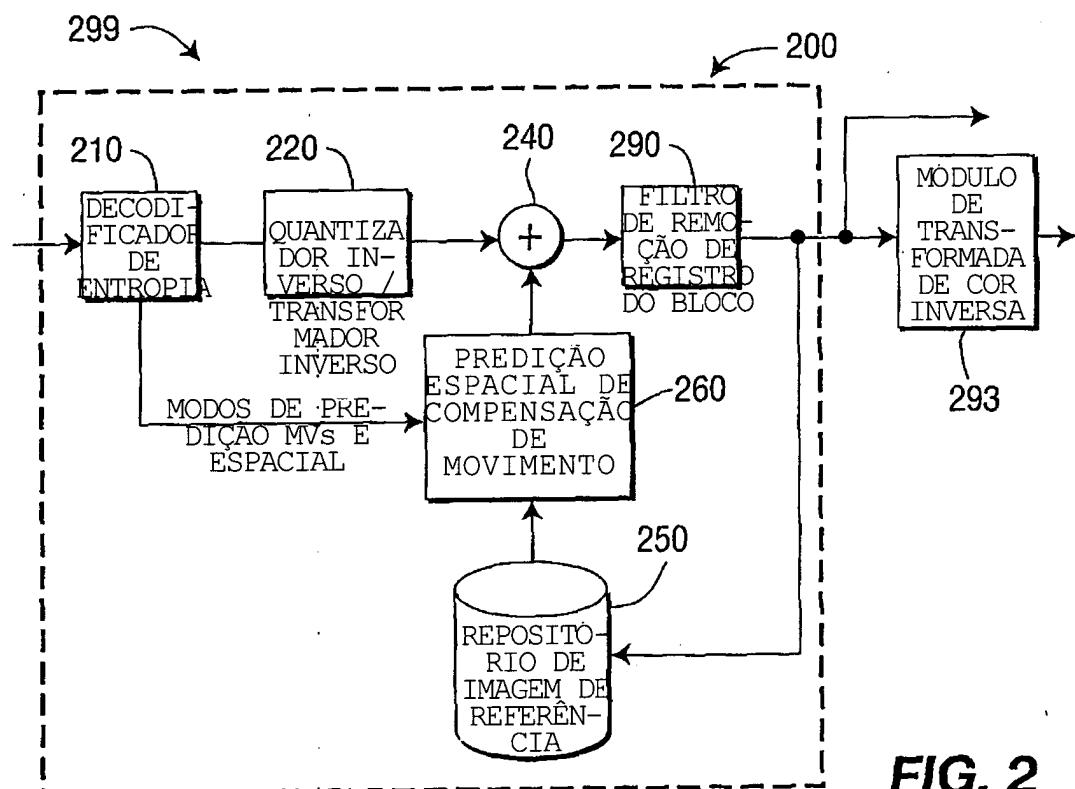
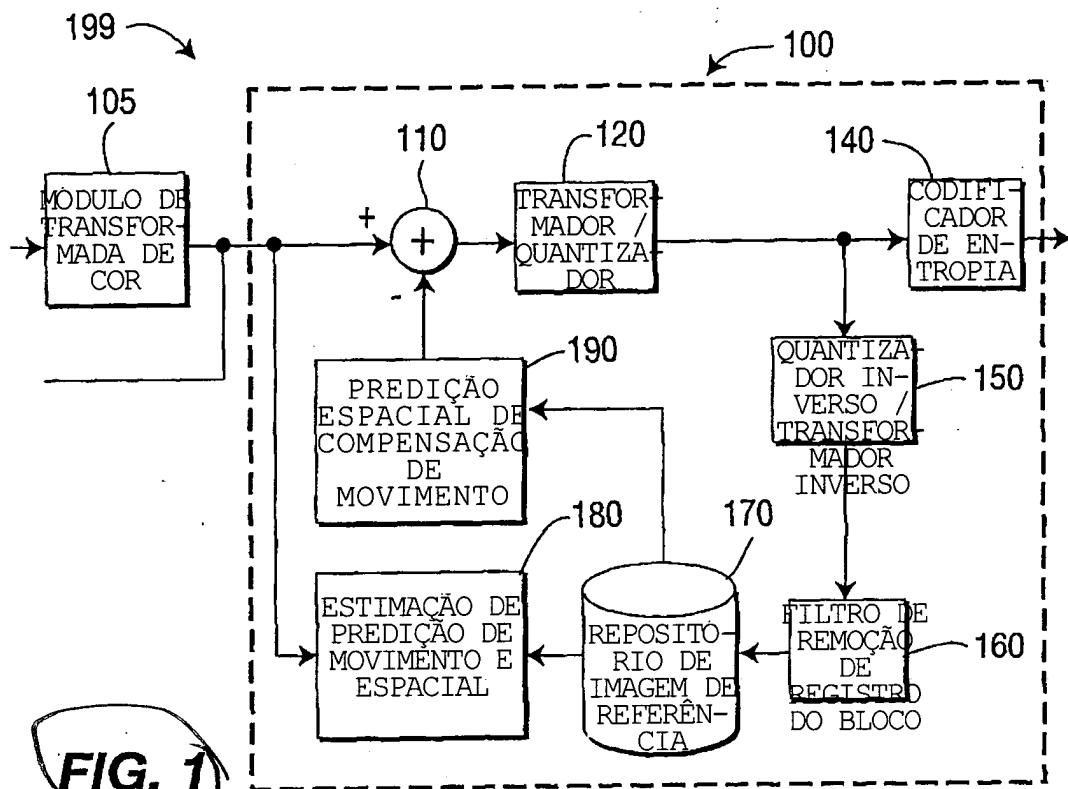
12. Método, de acordo com a reivindicação 8,

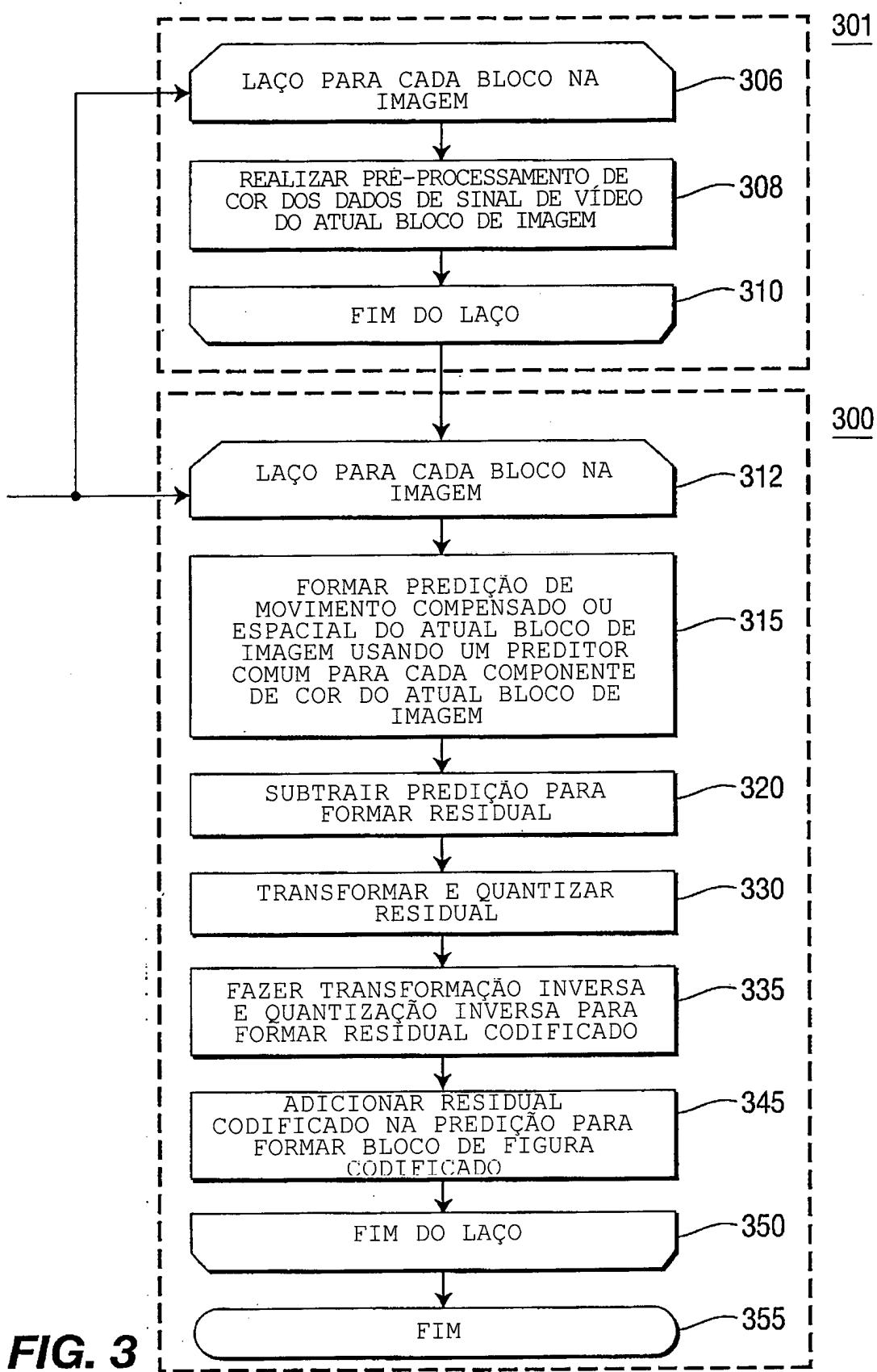
CARACTERIZADO pelo fato de que os filtros de interpolação comuns são usados para quadros B e P para todos os componentes de cor dos dados de sinal de vídeo.

13. Método, de acordo com a reivindicação 8,
5 **CARACTERIZADO** pelo fato de que a dita etapa de codificação codifica todos os componentes de cor dos dados de sinal de vídeo sem aplicar uma transformada de cor residual neles.

14. Método, de acordo com a reivindicação 8,
10 **CARACTERIZADO** pelo fato de que compreende adicionalmente realizar (308) uma transformada de cor nos dados de sinal de vídeo em uma etapa de pré-processamento antes da dita etapa de codificação.

15. Método, de acordo com a reivindicação 8,
15 **CARACTERIZADO** pelo fato de que a amostragem dos dados de sinal de vídeo corresponde a qualquer um dos formatos 4:4:4, 4:2:2 e 4:2:0 do padrão H.264 da União Internacional de Telecomunicações, Setor de Telecomunicações.





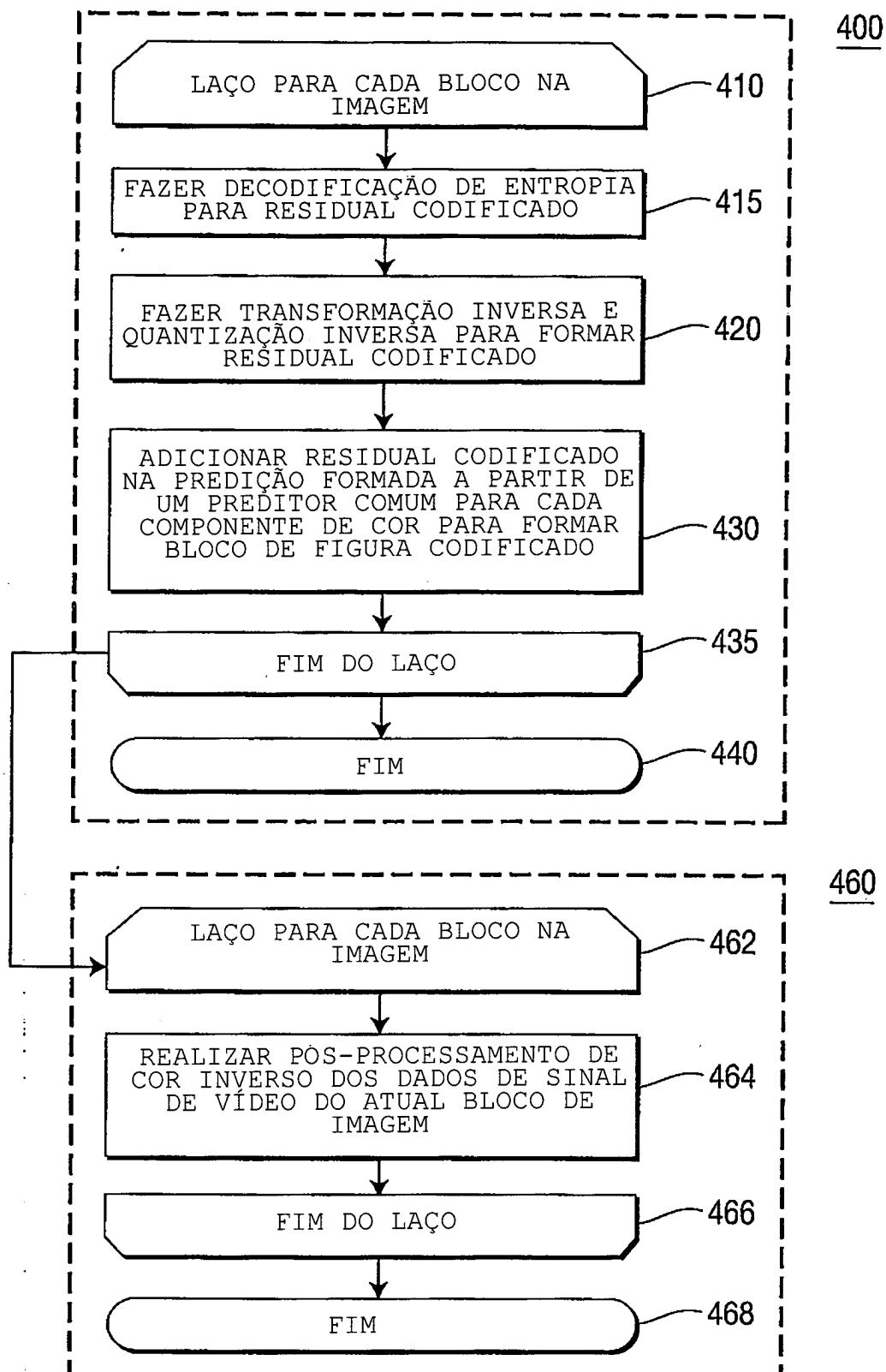


FIG. 4

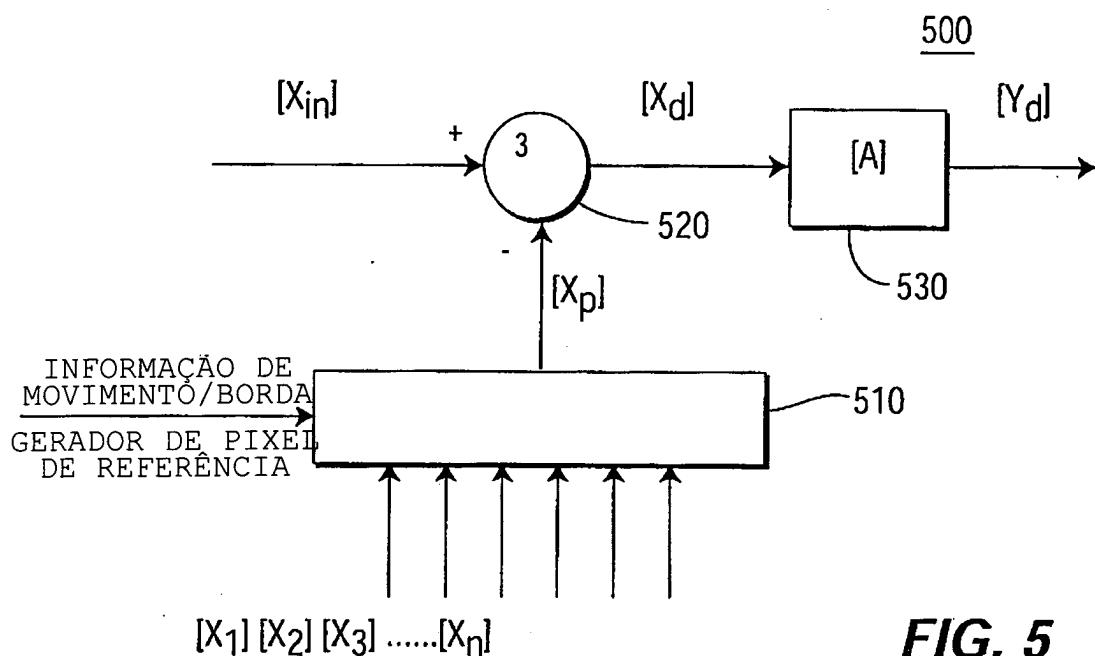
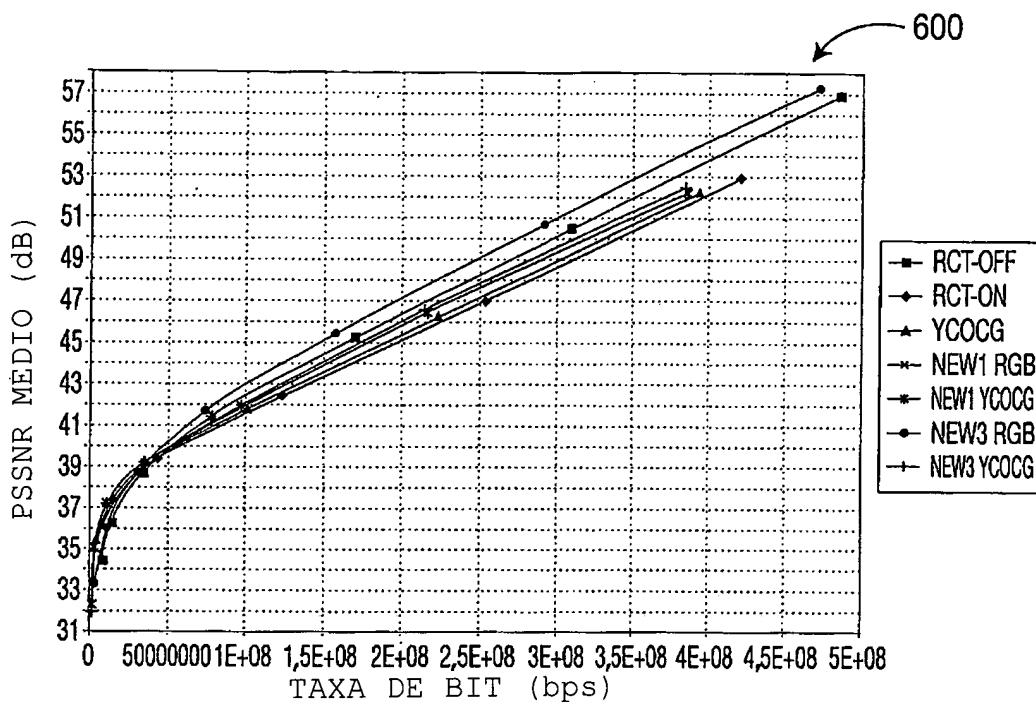
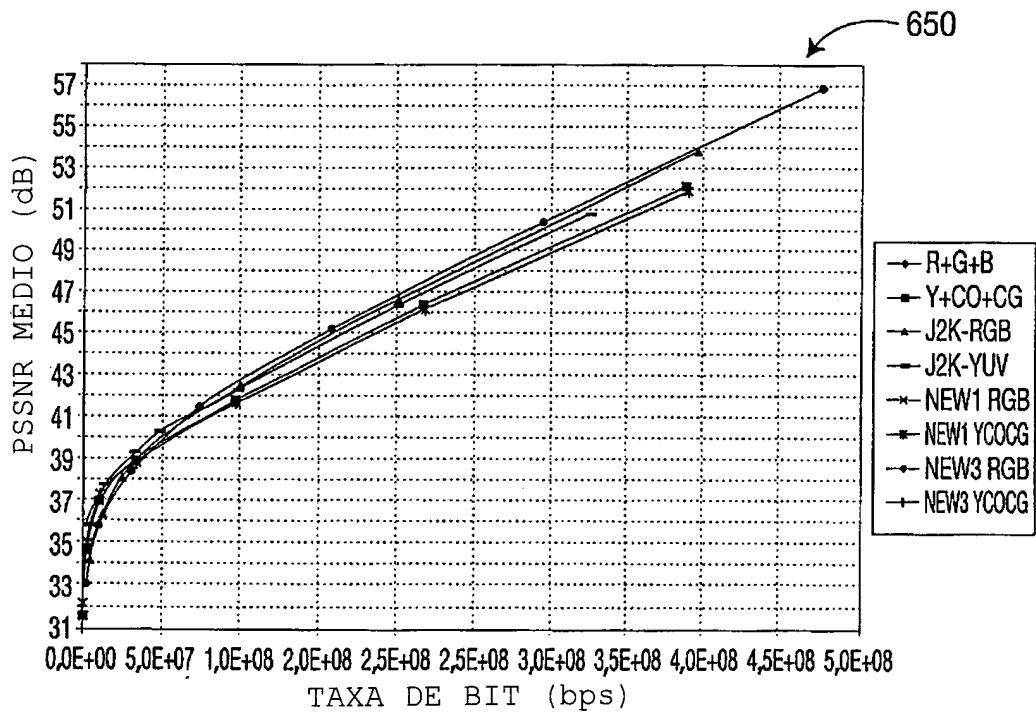


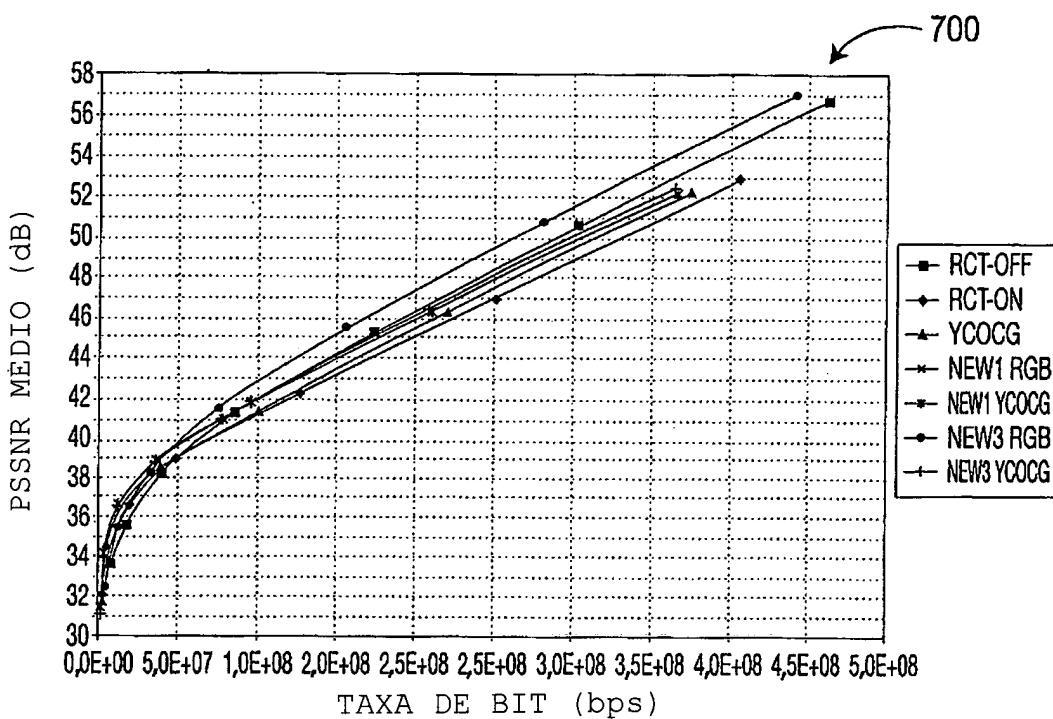
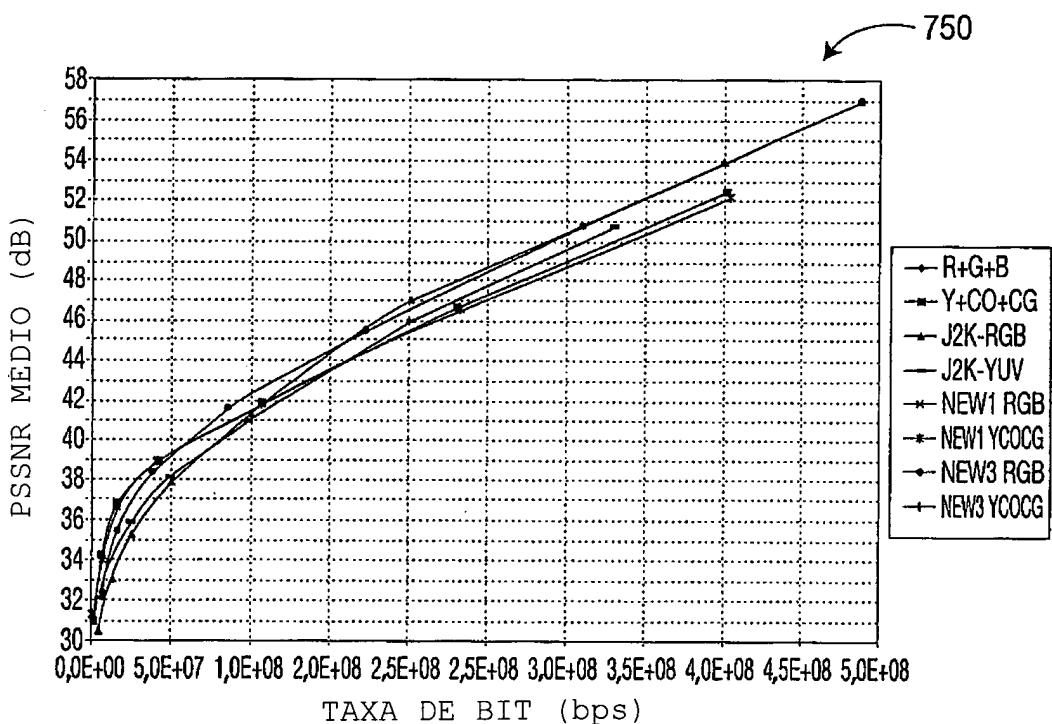
FIG. 5

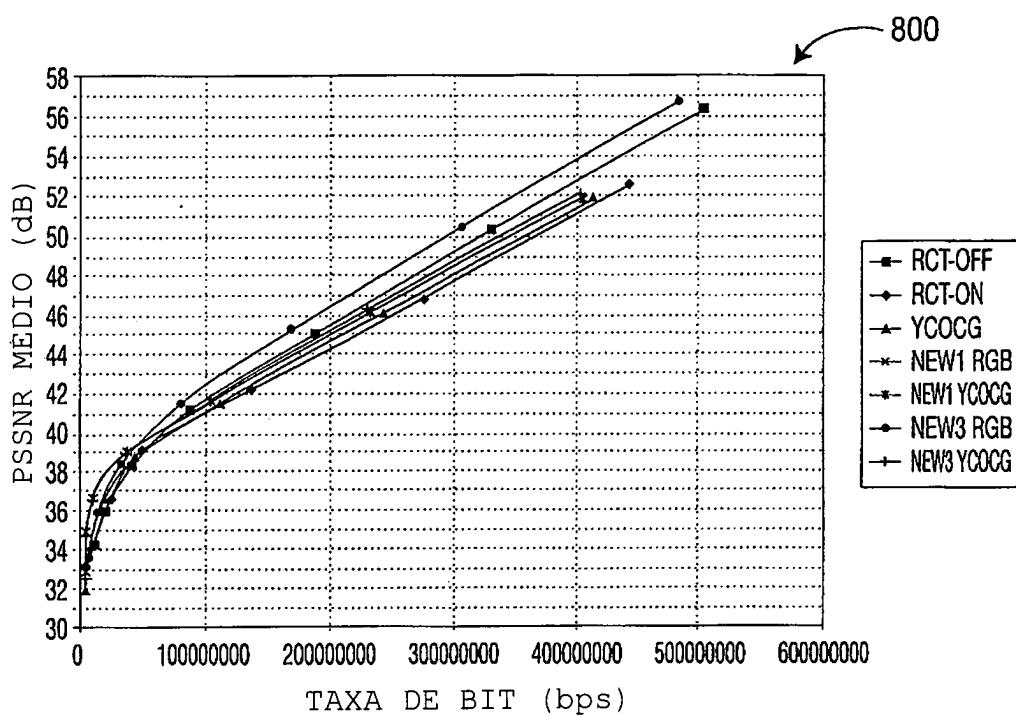
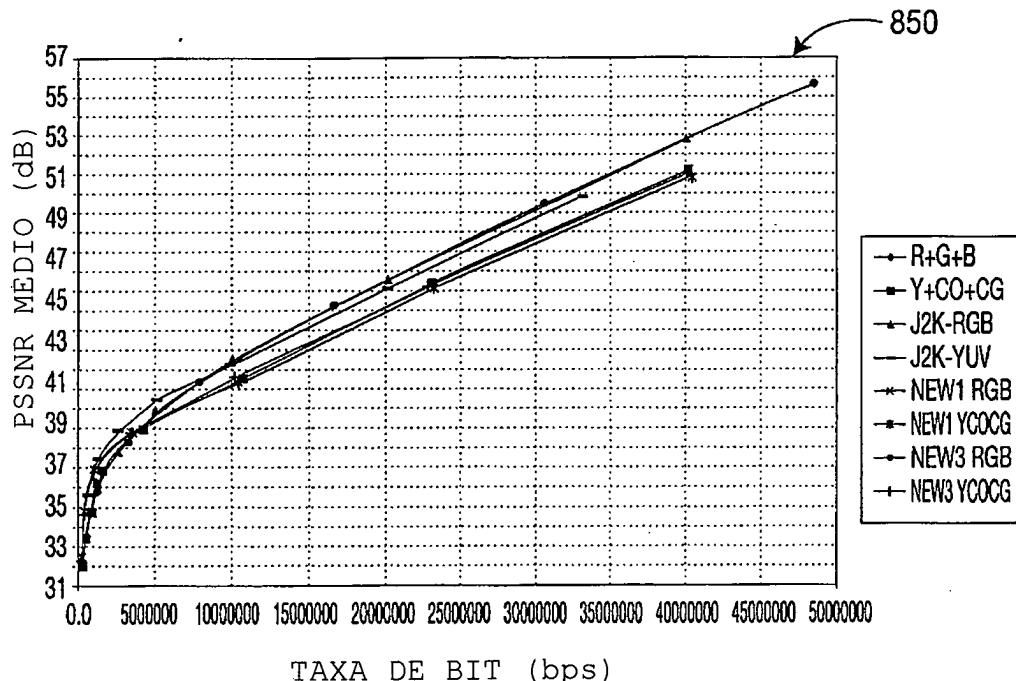
1200

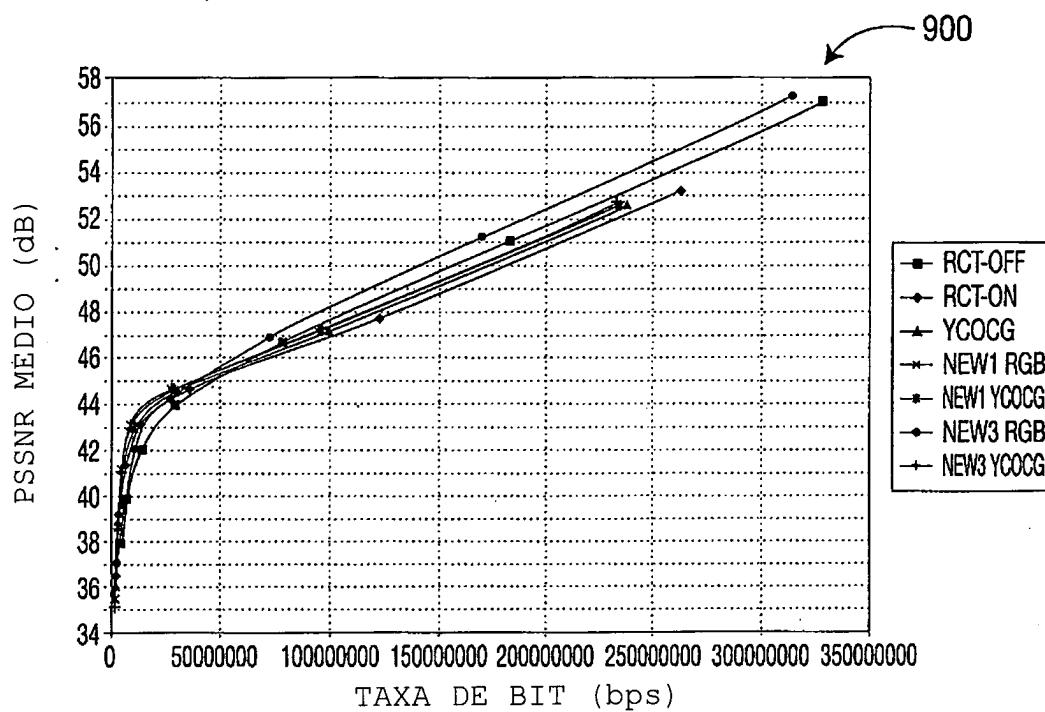
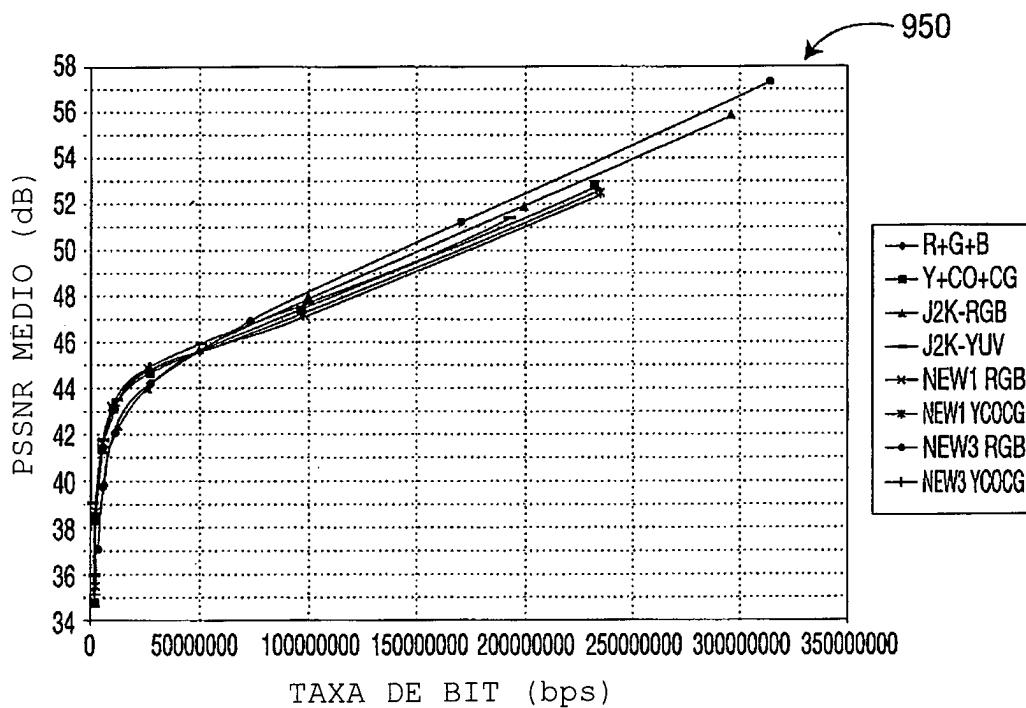
seq_parameter_set_rbsp(){	C	Descriptor
profile_idc	0	u(8)
constraint_set0_flag	0	u(1)
constraint_set1_flag	0	u(1)
constraint_set2_flag	0	u(1)
constraint_set3_flag	0	u(1)
reserved_zero_4bits /* equal to 0 */	0	u(4)
level_idc	0	u(8)
seq_parameter_set_id	0	ue(v)
if(profile_idc == 100 profile_idc == 110 profile_idc == 122 profile_idc == 166 profile_idc == 188){		
chroma_format_idc	0	ue(v)
bit_depth_luma_minus8	0	ue(v)

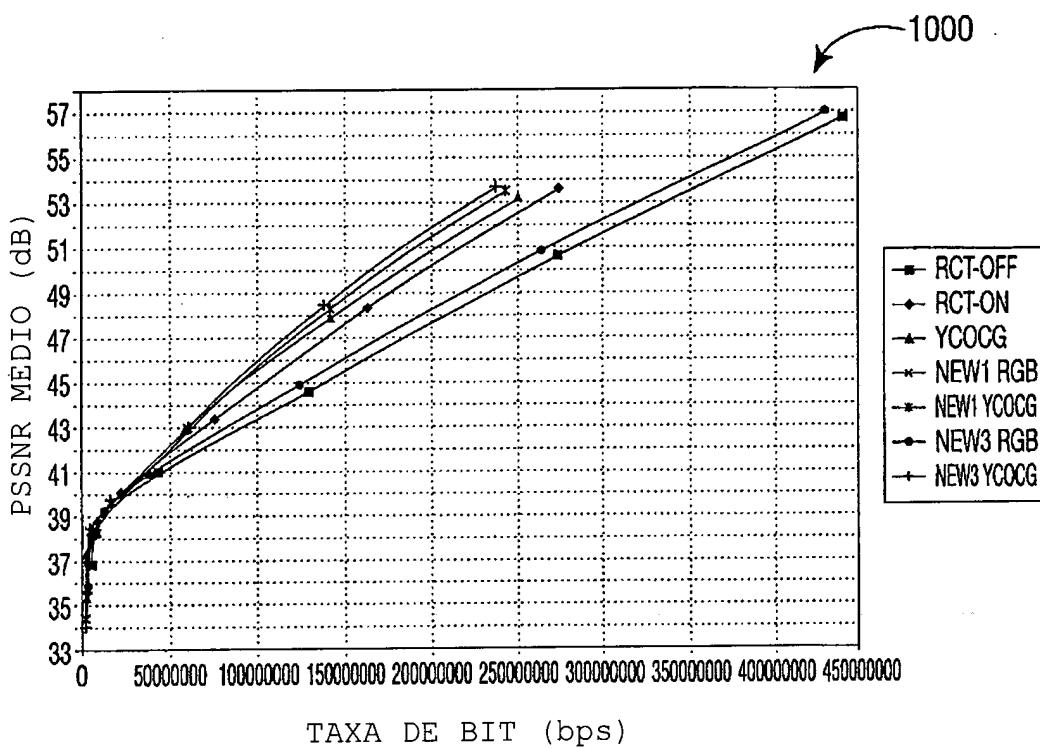
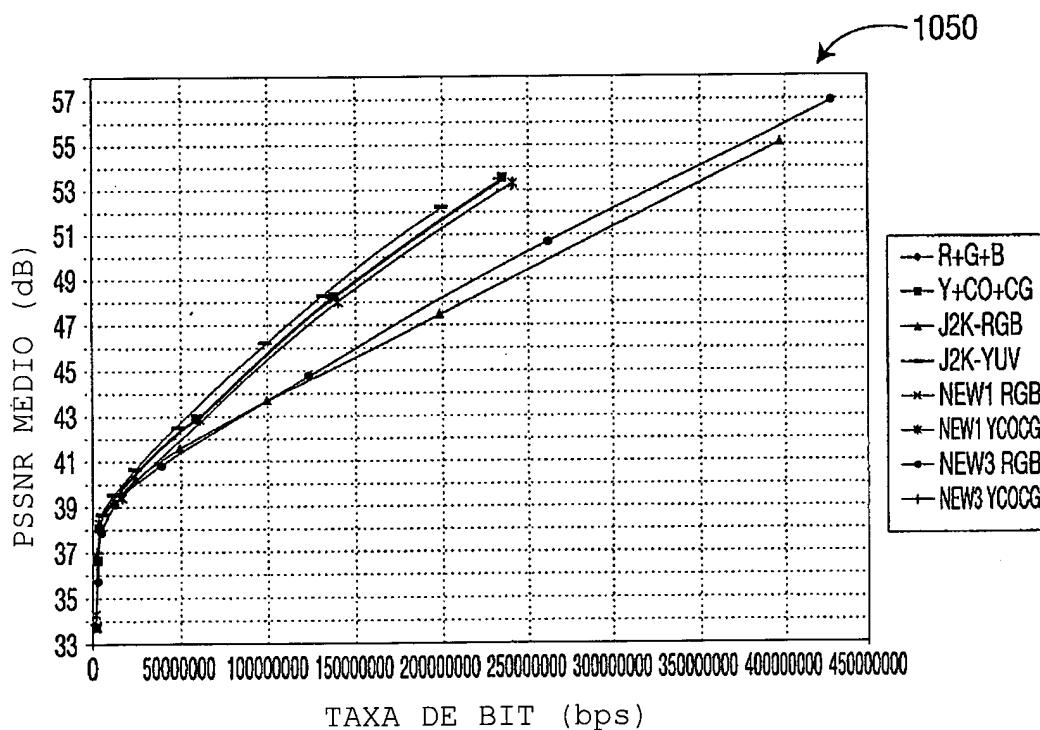
FIG. 12

**FIG. 6A****FIG. 6B**

**FIG. 7A****FIG. 7B**

**FIG. 8A****FIG. 8B**

**FIG. 9A****FIG. 9B**

**FIG. 10A****FIG. 10B**

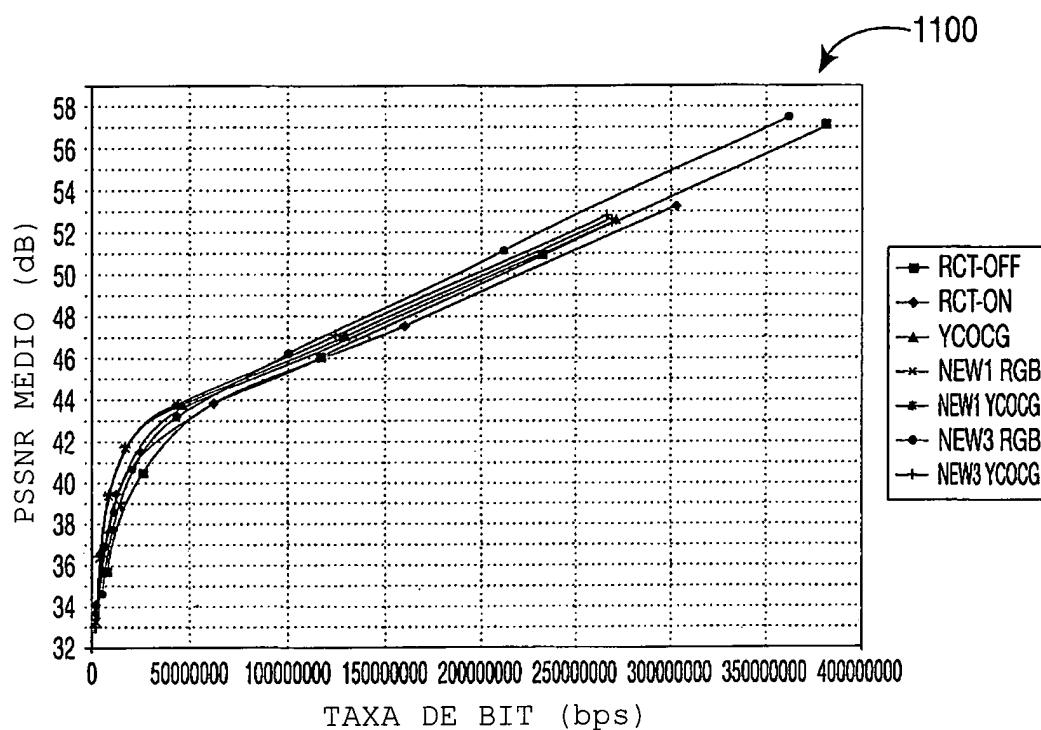
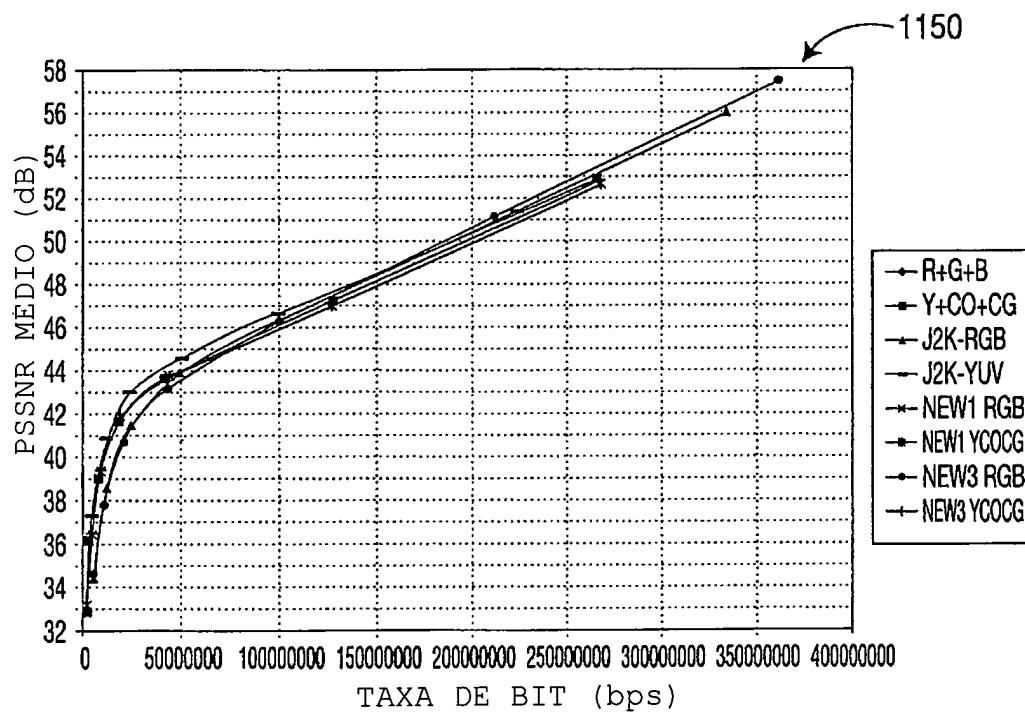
**FIG. 11A****FIG. 11B**

FIG. 13**FIG. 13A****FIG. 13B****FIG. 13C****FIG. 13D****FIG. 13A**

Residual()	C	Descriptor
if(!entropy_coding_mode_flag)		
residual_block = residual_block_cavlc		
else		
residual_block = residual_block_cabac		
if(MbPartPredMode(mb_type, 0) == Intra_16x16)		
residual_block(Intra16x16DCLevel, 16)	3	
for(i8x8 = 0; i8x8 < 4; i8x8++) /* each luma 8x8 block */		
if(!transform_size_8x8_flag !entropy_coding_mode_flag)		
for(i4x4 = 0; i4x4 < 4; i4x4++) { /* each 4x4 sub-block of block */		
if(CodedBlockPatternLuma & (1 << i8x8))		
if(MbPartPredMode(mb_type, 0) == Intra_16x16)		
residual_block(Intra16x16ACLevel[i8x8 * 4 + i4x4], 15)	3	
else		
residual_block(LumaLevel[i8x8 * 4 + i4x4], 16)	3 4	
else if(MbPartPredMode(mb_type, 0) == Intra_16x16)		
for(i = 0; i < 15; i++)		
Intra16x16ACLevel[i8x8 * 4 + i4x4][i] = 0		
else		
for(i = 0; i < 16; i++)		
LumaLevel[i8x8 * 4 + i4x4][i] = 0		
if(!entropy_coding_mode_flag && transform_size_8x8_flag)		
for(i = 0; i < 16; i++)		
LumaLevel8x8[i8x8][4 * i + i4x4] =		
LumaLevel[i8x8 * 4 + i4x4][i]		
}		

FIG. 13B

<pre> else if(CodedBlockPatternLuma & (1 << i8x8)) residual_block(LumaLevel8x8[i8x8], 64) else for(i=0; i<64; i++) LumaLevel8x8[i8x8][i]=0 if(chroma_format_idc == 1 chroma_format_idc==2){ NumC8x8=4 / (SubWidthC * SubHeightC) for(iCbCr = 0; iCbCr < 2; iCbCr++) if(CodedBlockPatternChroma & 3) /* chroma DC residual present */ residual_block(ChromaDCLevel[iCbCr], 4 * NumC8x8) else for(i=0; i< 4 * NumC8x8; i++) ChromaDCLevel[iCbCr][i]=0 for(iCbCr = 0; iCbCr < 2; iCbCr++) for(i8x8 = 0; i8x8 < NumC8x8; i8x8++) for(i4x4 = 0; i4x4 < 4; i4x4++) if(CodedBlockPatternChroma & 2) /* chroma AC residual present */ residual_block(ChromaACLevel[iCbCr][i8x8*4+i4x4],15) else for(i = 0; i < 15; i++) ChromaACLevel[iCbCr][i8x8*4+i4x4][i]=0 } else if(chroma_format_idc == 3){ if(MbPartPredMode(mb_type, 0) == Intra_16x16)/*Cb for 4:4:4*/ residual_block(CbIntra16x16DCLevel, 16) for(i8x8 = 0; i8x8 < 4; i8x8++) /* each Cb 8x8 block */ if(!transform_size_8x8_flag !entropy_coding_mode_flag) for(i4x4 = 0; i4x4 < 4; i4x4++) { /* each 4x4 sub-block of block */ if(CodedBlockPatternLuma & (1 << i8x8)) </pre>	3 4
	3 4
	3 4
	3

FIG. 13C

<pre> if(MbPartPredMode(mb_type, 0) == Intra_16x16) residual_block(CbIntra16x16ACLevel[i8x8*4+i4x4],15) else residual_block(CbLevel[i8x8 * 4 + i4x4], 16) else if(MbPartPredMode(mb_type, 0) == Intra_16x16) for(i = 0; i < 15; i++) CbIntra16x16ACLevel[i8x8 * 4 + i4x4][i] = 0 else for(i = 0; i < 16; i++) CbLevel[i8x8 * 4 + i4x4][i] = 0 if(!entropy_coding_mode_flag && transform_size_8x8_flag) for(i = 0; i < 16; i++) CbLevel8x8[i8x8][4 * i + i4x4] = CbLevel[i8x8 * 4 + i4x4][i] } else if(CodedBlockPatternLuma & (1 << i8x8)) residual_block(CbLevel8x8[i8x8], 64) else for(i = 0; i < 64; i++) CbLevel8x8[i8x8][i] = 0 if(MbPartPredMode(mb_type, 0) == Intra_16x16) /*Cr for 4:4:4 */ residual_block(CrIntra16x16DCLevel, 16) for(i8x8 = 0; i8x8 < 4; i8x8++) /* each Cr 8x8 block */ if(!transform_size_8x8_flag !entropy_coding_mode_flag) for(i4x4 = 0; i4x4 < 4; i4x4++) { /* each 4x4 sub-block of block */ if(CodedBlockPatternLuma & (1 << i8x8)) if(MbPartPredMode(mb_type, 0) == Intra_16x16) residual_block(CrIntra16x16ACLevel[i8x8*4+i4x4],15) else </pre>	3
	3 4
	3 4
	3
	3

FIG. 13D

```

    residual_block( CrLevel[ i8x8 * 4 + i4x4 ], 16 )
else if( MbPartPredMode( mb_type, 0 ) == Intra_16x16 )
    for( i=0; i<15; i++ )
        CrIntra16x16ACLevel[ i8x8 * 4 + i4x4 ][ i ]=0
else
    for( i=0; i<16; i++ )
        CrLevel[ i8x8 * 4 + i4x4 ][ i ]=0
if( !entropy_coding_mode_flag && transform_size_8x8_flag )
    for( i=0; i<16; i++ )
        CrLevel8x8[ i8x8 ][ 4 * i + i4x4 ]=
            CrLevel[ i8x8 * 4 + i4x4 ][ i ]
    }
else if( CodedBlockPatternLuma & ( 1 << i8x8 ) )
    residual_block( CrLevel8x8[ i8x8 ], 64 )
else
    for( i=0; i<64; i++ )
        CrLevel8x8[ i8x8 ][ i ]=0
}

```

3 | 4

3 | 4

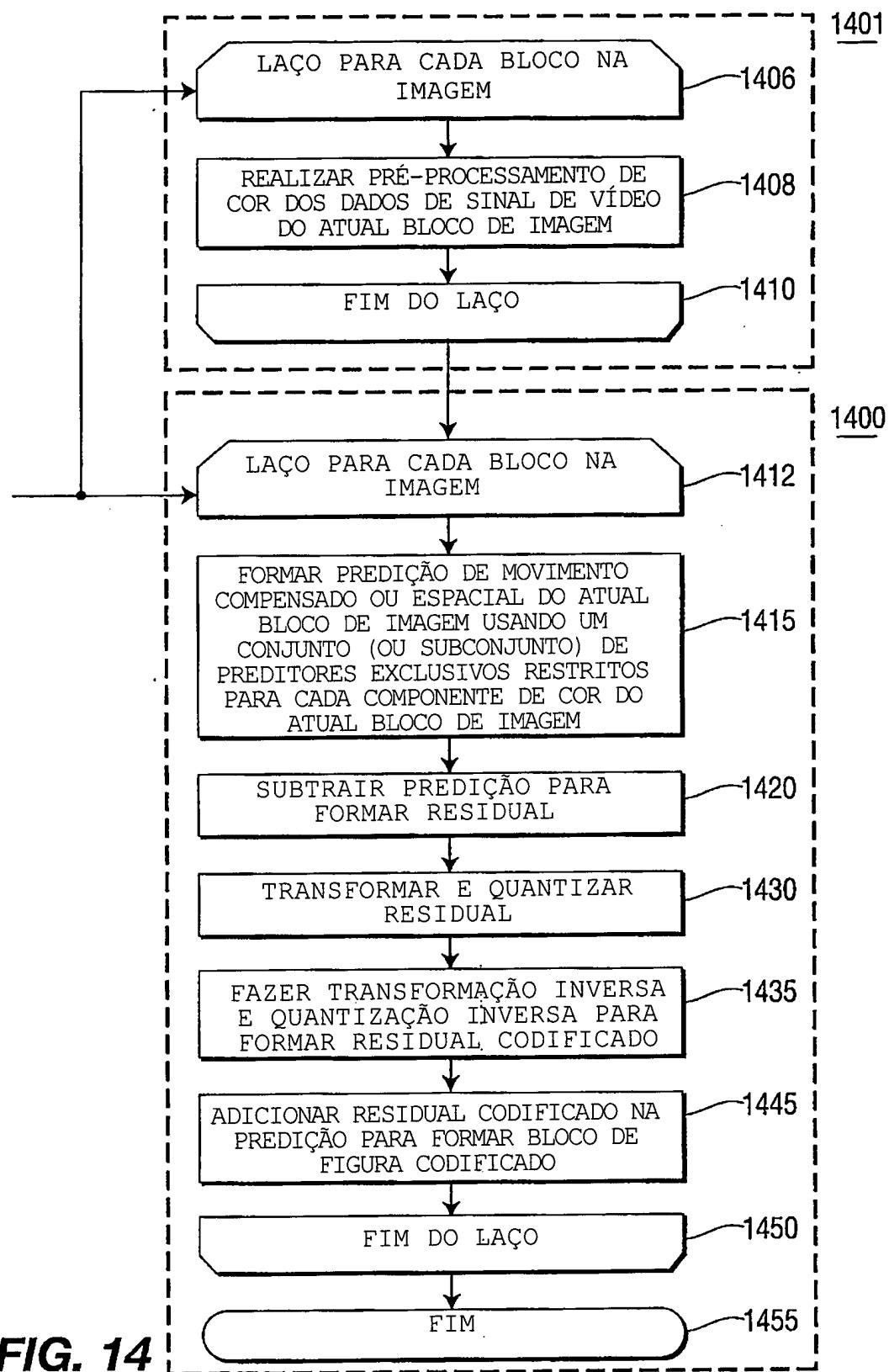
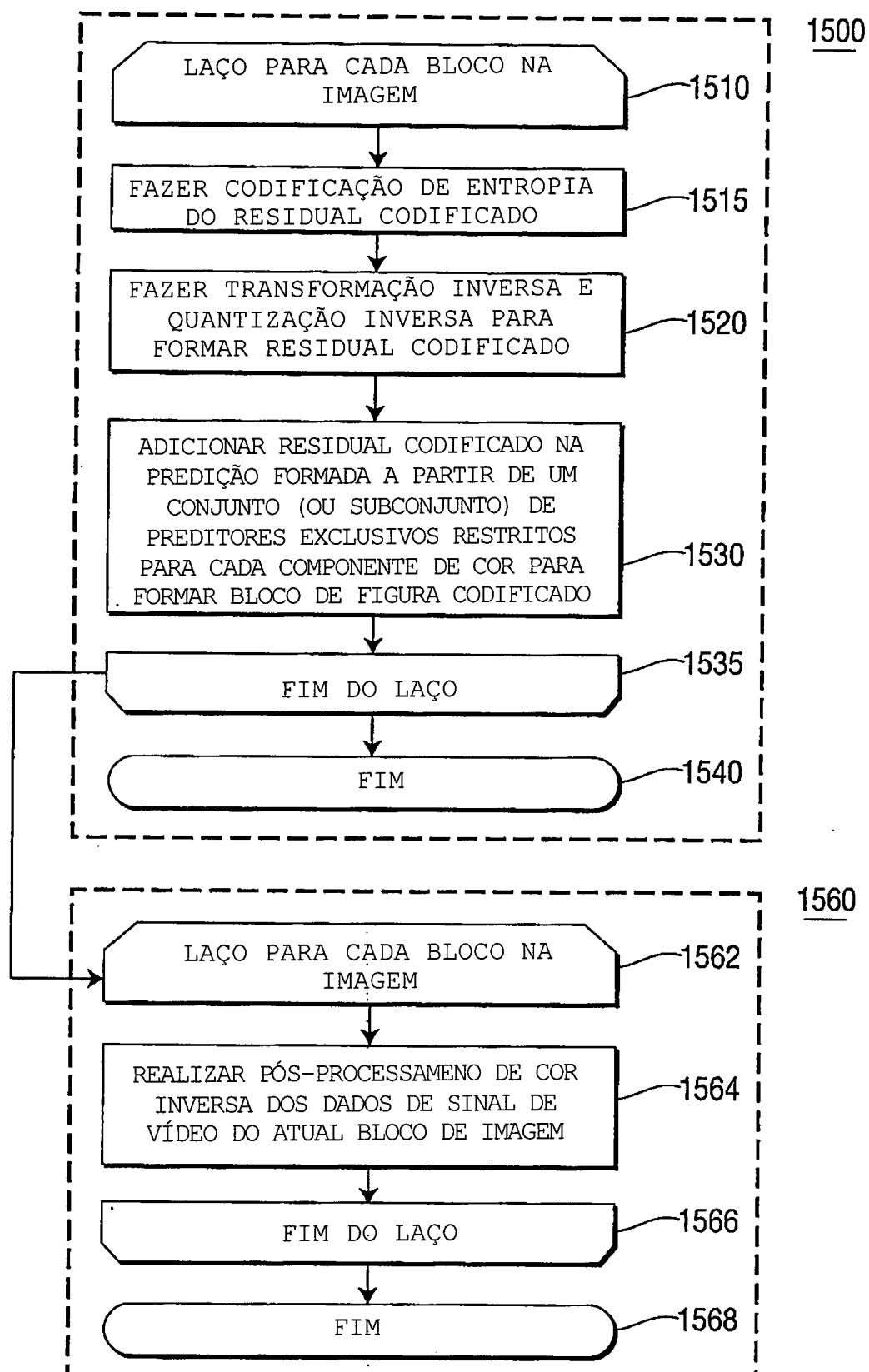


FIG. 14

**FIG. 15**

1600

FIG. 16A**FIG. 16****FIG. 16A****FIG. 16B**

mb_pred(mb_type){	C	Descriptor
if(MbPartPredMode(mb_type, 0) == Intra_4x4	2	u(1) ae(v)
MbPartPredMode(mb_type, 0) == Intra_8x8	2	u(3) ae(v)
MbPartPredMode(mb_type, 0) == Intra_16x16){		
if(profile_idc != 166) {		
if(MbPartPredMode(mb_type, 0) == Intra_4x4)		
for(luma4x4BlkIdx=0; luma4x4BlkIdx<16; luma4x4BlkIdx++) {		
prev_intra4x4_pred_mode_flag[luma4x4BlkIdx]		
if(!prev_intra4x4_pred_mode_flag[luma4x4BlkIdx])		
rem_intra4x4_pred_mode[luma4x4BlkIdx]		
}		
if(MbPartPredMode(mb_type, 0) == Intra_8x8)		
for(luma8x8BlkIdx=0; luma8x8BlkIdx<4; luma8x8BlkIdx++) {		
prev_intra8x8_pred_mode_flag[luma8x8BlkIdx]		
if(!prev_intra8x8_pred_mode_flag[luma8x8BlkIdx])		
rem_intra8x8_pred_mode[luma8x8BlkIdx]		
}		
if(chroma_format_idc != 0)		
intra_chroma_pred_mode		
} else {		
if(MbPartPredMode(mb_type, 0) == Intra_4x4)		
for(luma4x4BlkIdx=0; luma4x4BlkIdx<16; luma4x4BlkIdx++) {		
prev_intra4x4_pred_mode_flag0[luma4x4BlkIdx]		
if(!prev_intra4x4_pred_mode_flag0[luma4x4BlkIdx])		
rem_intra4x4_pred_mode0[luma4x4BlkIdx]		
prev_intra4x4_pred_mode_flag1[luma4x4BlkIdx]		

FIG. 16B

if(!prev_intra4x4_pred_mode_flag1[luma4x4BlkIdx])	2	u(3) ae(v)
rem_intra4x4_pred_mode1[luma4x4BlkIdx]	2	u(1) ae(v)
prev_intra4x4_pred_mode_flag2[luma4x4BlkIdx]	2	u(3) ae(v)
if(!prev_intra4x4_pred_mode_flag2[luma4x4BlkIdx])	2	u(3) ae(v)
rem_intra4x4_pred_mode2[luma4x4BlkIdx]	2	u(1) ae(v)
}		
if(MbPartPredMode(mb_type, 0) == Intra_8x8)		
for(luma8x8BlkIdx=0; luma8x8BlkIdx<4; luma8x8BlkIdx++) {		
prev_intra8x8_pred_mode_flag0[luma8x8BlkIdx]	2	u(1) ae(v)
if(!prev_intra8x8_pred_mode_flag0[luma8x8BlkIdx])	2	u(3) ae(v)
rem_intra8x8_pred_mode0[luma8x8BlkIdx]	2	u(1) ae(v)
prev_intra8x8_pred_mode_flag1[luma8x8BlkIdx]	2	u(3) ae(v)
if(!prev_intra8x8_pred_mode_flag1[luma8x8BlkIdx])	2	u(1) ae(v)
rem_intra8x8_pred_mode1[luma8x8BlkIdx]	2	u(3) ae(v)
prev_intra8x8_pred_mode_flag2[luma8x8BlkIdx]	2	u(1) ae(v)
if(!prev_intra8x8_pred_mode_flag2[luma8x8BlkIdx])	2	u(3) ae(v)
rem_intra8x8_pred_mode2[luma8x8BlkIdx]	2	u(3) ae(v)
}		
}		
} else if(MbPartPredMode(mb_type, 0) != Direct) {		
.....		

RESUMO

"CODIFICAÇÃO LUMA E CROMA USANDO UM PREDITOR COMUM"

São descritos codificadores de vídeo, decodificadores de vídeo e métodos correspondentes. Um codificador de vídeo para codificar dados de sinal de vídeo para um bloco de imagem inclui um codificador (100) para codificar todos os componentes de cor dos dados de sinal de vídeo usando um preditor comum (315). Um decodificador de vídeo para decodificar dados de sinal de vídeo para um bloco de imagem inclui um decodificador (200) para decodificar todos os componentes de cor dos dados de sinal de vídeo usando um preditor comum (430). Adicionalmente, um aparelho e método para codificar e decodificar dados de sinal para um bloco de imagem incluem 15 um codificador e decodificador para codificar / decodificar componentes de cor dos dados de sinal de vídeo sem aplicar uma transformada de cor residual neles. Além do mais, um codificador e decodificador de vídeo para codificar / decodificar dados de sinal de vídeo para um bloco de imagem incluem 20 um codificador e decodificador para codificar / decodificar os dados de sinal de vídeo usando preditores exclusivos para cada um dos componentes de cor dos dados de sinal de vídeo.

REIVINDICAÇÕES

1. Codificador de vídeo para codificar dados de sinal de vídeo para um bloco de imagem, **CARACTERIZADO** pelo fato de que o codificador de vídeo comprehende um codificador 5 (100) para codificar todos os componentes de cor dos dados de sinal de vídeo usando um preditor comum.

2. Codificador de vídeo, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o preditor comum é um preditor de luminância usado tanto por componentes de 10 luminância quanto por componentes de crominância dos dados de sinal de vídeo.

3. Codificador de vídeo, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o dito codificador 15 (100) usa um modo de predição espacial comum para todos os componentes de cor dos dados de sinal de vídeo.

4. Codificador de vídeo, de acordo com a reivindicação 3, **CARACTERIZADO** pelo fato de que o modo de predição espacial comum é ajustado pelos parâmetros `prev_intra8x8_pred_mode_flag`, `rem_intra8x8_pred_mode`, 20 `prev_intra4x4_pred_mode_flag` e `rem_intra4x4_pred_mode` do padrão H.264 da União Internacional de Telecomunicações, Setor de Telecomunicações.

5. Codificador de vídeo, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o dito codificador 25 (100) usa filtros de interpolação comuns para quadros B e P para todos os componentes de cor dos dados de sinal de vídeo.

6. Codificador de vídeo, de acordo com a reivindi-

cação 1, **CARACTERIZADO** pelo fato de que o dito codificador (100) codifica todos os componentes de cor dos dados de sinal de vídeo sem aplicar uma transformada de cor residual neles.

7. Codificador de vídeo, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que a amostragem dos dados de sinal de vídeo corresponde a qualquer um dos formatos 4:4:4, 4:2:2 e 4:2:0 do padrão H.264 da União Internacional de Telecomunicações, Setor de Telecomunicações.

8. Método para codificar dados de sinal de vídeo para um bloco de imagem, **CARACTERIZADO** pelo fato de que o método compreende codificar (315) todos os componentes de cor dos dados de sinal de vídeo usando um preditor comum.

9. Método, de acordo com a reivindicação 8, **CARACTERIZADO** pelo fato de que o preditor comum é um preditor de luminância usado tanto para componentes de luminância quanto para componentes de crominância dos dados de sinal de vídeo.

10. Método, de acordo com a reivindicação 8, **CARACTERIZADO** pelo fato de que um modo de predição espacial comum é usado para todos os componentes de cor dos dados de sinal de vídeo.

11. Método, de acordo com a reivindicação 10, **CARACTERIZADO** pelo fato de que o modo de predição espacial comum é ajustado pelos parâmetros prev_intra8x8_pred_mode_flag, rem_intra8x8_pred_mode, prev_intra4x4_pred_mode_flag e rem_intra4x4_pred_mode do padrão H.264 da União Internacional de Telecomunicações, Setor de Telecomunicações.

12. Método, de acordo com a reivindicação 8,