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

(21) PI 0622050-9 A2

(22) Data de Depósito: 28/09/2006
(43) Data da Publicação: 22/04/2014
(RPI 2259)

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

(54) Título: CODIFICAÇÃO DE REDUNDÂNCIA FLEXÍVEL

(57) Resumo:

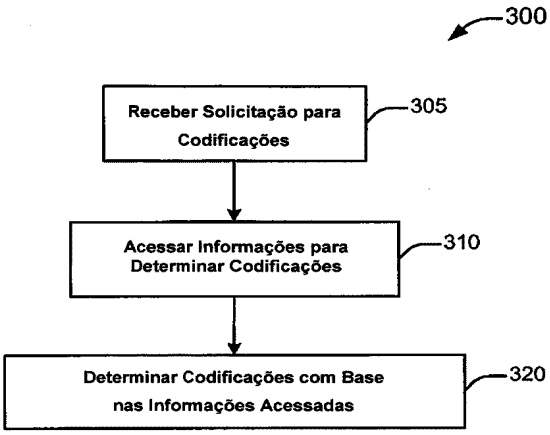
(73) Titular(es): Thomson Licensing

(72) Inventor(es): Jill Macdonald Boyce, Zhenyu Wu

(74) Procurador(es): ISABELLA CARDOZO

(86) Pedido Internacional: PCT US2006038184 de 28/09/2006

(87) Publicação Internacional: WO 2008/039201de 03/04/2008



"CODIFICAÇÃO DE REDUNDÂNCIA FLEXÍVEL"

Campo técnico

Essa revelação se refere à codificação de dados.

Antecedentes da invenção

Sistemas de codificação fornecem, frequentemente, redundância de modo que dados transmitidos podem ser recebidos e decodificados apesar da presença de erros. Sistemas específicos fornecem, no contexto de vídeo por exemplo, múltiplas codificações para uma sequência específica de imagens. Esses sistemas também transmitem todas as múltiplas codificações. Um receptor que recebe as codificações transmitidas pode ser capaz de utilizar codificações redundantes para decodificar corretamente a sequência específica mesmo se uma ou mais das codificações for perdida ou recebida com erros.

Sumário

De acordo com uma implementação, informações são acessadas para determinar qual das múltiplas codificações de pelo menos uma porção de um objeto de dados a enviar através de um canal, e um conjunto de codificações a enviar através do canal é determinado. O conjunto é determinado a partir das múltiplas codificações e inclui pelo menos uma e possivelmente mais de uma das múltiplas codificações. O número de codificações no conjunto determinado se baseia nas informações acessadas.

De acordo com outra implementação, informações são fornecidas para determinar qual das múltiplas codificações de pelo menos uma porção de um objeto de dados a enviar através de um canal. Um conjunto de codificações é recebido através do canal, com o conjunto tendo sido determinado a partir das múltiplas codificações e incluindo pelo menos uma e possivelmente mais de uma das múltiplas codificações. O número de codificações no conjunto se baseia nas informações fornecidas.

Os detalhes de uma ou mais implementações são expostos nos desenhos em anexo e descrição abaixo. Outros aspectos e características tornar-se-ão evidentes a partir da seguinte descrição detalhada considerada em combinação com os desenhos em anexo e reivindicações. Deve ser entendido, entretanto, que os desenhos são projetados exclusivamente para fins de ilustração e não como definição dos limites dos presentes princípios. Deve ser adicionalmente entendido que os desenhos não são necessariamente traçados em escala e que, a menos que de outro modo indicado, são meramente destinados a ilustrar de forma conceptual estruturas e procedimentos específicos.

BREVE DESCRIÇÃO DOS DESENHOS

A figura 1 inclui um diagrama de blocos de um sistema para enviar e receber dados codificados.

A figura 2 inclui um diagrama de blocos de outro sistema para enviar e receber dados codificados.

A figura 3 inclui um fluxograma de um processo para selecionar codificações com os sistemas das figuras 1 e 2.

A figura 4 inclui um fluxograma de um processo para receber codificações com os sistemas das figuras 1 e 2.

A figura 5 inclui um fluxograma de um processo para enviar codificações com o sistema da figura 2.

A figura 6 inclui uma representação pictorial de múltiplas codificações para cada uma de N imagens.

A figura 7 inclui uma representação pictorial de codificações selecionadas a partir da representação da figura 6.

A figura 8 inclui um fluxograma de um processo para processar codificações recebidas com o sistema da figura 2.

A figura 9 inclui um diagrama de blocos de um sistema para enviar e receber dados codificados utilizando camadas.

A figura 10 inclui um fluxograma de um processo para enviar codificações com o sistema da figura 9.

A figura 11 inclui uma representação pictorial de codificações de dados da figura 6 ordenadas em camadas de acordo com o processo da figura 10.

Descrição detalhada

Uma implementação é dirigida à codificação de vídeo utilizando o padrão H.264/AVC (Codificação avançada de vídeo) promulgada pelos órgãos de padrões ISO ("International Standards Organization") e MPEG ("Moving Picture Experts Group"). O padrão H.264/AVC descreve uma característica de "fatia redundante" que permite que uma imagem específica, por exemplo, seja codificada múltiplas vezes, desse modo fornecendo redundância. Utilizando a característica de "fatia redundante", a imagem específica pode ser codificada uma primeira vez como uma "imagem codificada primária" ("PCP") e uma ou mais vezes adicionais como uma ou mais "imagens codificadas redundantes" ("RCPs"). Uma imagem codificada, quer seja uma PCP ou uma RCP, pode incluir múltiplas fatias, porém para fins de simplicidade os requerentes utilizam, tipicamente, esses termos de forma intercambiável, como se a imagem codificada incluísse somente uma única fatia.

A implementação acima codifica a imagem específica antecipadamente criando uma PCP bem como múltiplas RCPs. Quando uma transmissão da imagem específica é solicitada, por exemplo, por um usuário que solicita um download através da Internet, um transmissor acessa essas imagens codificadas. O transmissor também acessa informações que descrevem, por exemplo, a taxa atual de erro no percurso até o usuário. Com base na taxa atual de erro, o transmissor determina qual das múltiplas RCPs enviar para o usuário, juntamente com a PCP. O transmissor pode determinar, por exemplo, o envio somente de

uma RCP se a taxa de erro for baixa, porém enviar todas as múltiplas RCPs se a taxa de erro for elevada.

A figura 1 mostra um diagrama de blocos de um sistema 100 para enviar e receber dados codificados. O sistema 100 inclui uma fonte de codificação 110 que fornece codificações através de um percurso 120 para um compilador 130. O compilador 130 recebe informações a partir de uma fonte de informações 140, e provê um fluxo compilado de codificações através de um percurso 150 para um dispositivo de armazenagem/receptor 160.

O sistema 100 pode ser implementado utilizando qualquer um de uma variedade de diferentes padrões ou métodos de codificação, e não necessita estar em conformidade com qualquer padrão. Por exemplo, a fonte 110 pode ser um computador pessoal ou outro dispositivo de computação que codifica dados utilizando várias técnicas de codificação por estimação de movimento diferentes, ou mesmo códigos de bloco. A fonte 110 também pode ser, por exemplo, um dispositivo de armazenagem que armazena codificações que foram codificadas por um tal dispositivo de computação. Entretanto, para clareza e perfeição em descrição, grande parte desse pedido descreve implementações específicas que utilizam padrão de codificação H.264/AVC. Apesar dos detalhes e foco dessas implementações no padrão H.264/AVC, outras implementações são consideradas que não utilizam nenhum padrão, muito menos o padrão H.264/AVC.

O compilador 130 recebe da fonte 110 múltiplas codificações para uma dada unidade de dados. O compilador 130 seleciona pelo menos algumas das múltiplas codificações a enviar para o dispositivo de armazenagem/receptor 160, e compila as codificações selecionadas para enviar as codificações selecionadas para o dispositivo de armazenagem/receptor 160. Em muitas implementações, o compilador 130 compila e envia codificações em resposta a uma solicitação, ou após receber uma solicitação.

Uma tal solicitação pode ser recebida, por exemplo, da fonte 110, do dispositivo de armazenagem/receptor 160, ou de outro dispositivo não mostrado no sistema 100. Tais outros dispositivos podem incluir, por exemplo, um servidor de rede listando codificações disponíveis na fonte 110 e fornecendo acesso dos usuários às codificações listadas. Em uma tal implementação, o servidor de rede pode conectar-se ao compilador 130 para solicitar que codificações sejam enviadas para o dispositivo de armazenagem/receptor 160 onde um usuário pode estar fisicamente localizado. O dispositivo de armazenagem/receptor 160 pode fornecer ao usuário, por exemplo, um meio de exibição de alta definição para visualizar codificações (por exemplo, vídeos) que são recebidos, e um browser para selecionar vídeos a partir do servidor de rede.

O compilador 130 também pode compilar e enviar codificações sem uma solicitação. Por exemplo, o compilador 130 pode simplesmente compilar e enviar codificações em resposta ao recebimento de um fluxo de codificações a partir da fonte 110. Como exemplo

adicional, o compilador 130 pode compilar e enviar codificações em um tempo fixo toda noite para fornecer um fluxo compilado diariamente dos eventos de notícias do dia, e o fluxo pode ser empurrado para uma variedade de receptores.

O compilador 130 baseia a seleção de codificações a compilar e enviar, pelo menos em parte, em informações recebidas da fonte de informações 140. As informações recebidas podem relacionar-se a um ou mais de vários fatores incluindo, por exemplo, (1) qualidade de serviço, ou um tipo de serviço, esperado ou desejado para a dada unidade de dados, (2), capacidade (bits ou largura de banda, por exemplo) alocada a dada unidade de dados, (3) taxa de erro (taxa de erro de bits ou taxa de erro de pacote, por exemplo) no percurso (também mencionado como canal) para o dispositivo de armazenagem/receptor 160, e (4) capacidade disponível no percurso para o dispositivo de armazenagem/receptor 160. Muitos fatores referem-se a condições de canal (também mencionado como desempenho de canal), como, por exemplo, capacidade ou taxa de erro. A fonte de informações 140 pode ser, por exemplo, (1) uma unidade de controle que monitora o percurso 150, (2) um gerenciador de qualidade de serviço que pode ser, por exemplo, local para o compilador 130, ou (3) uma tabela de consulta incluída no compilador 130 que fornece taxas de bit alvo para várias unidades de dados.

O compilador 130 pode utilizar as informações a partir da fonte de informações 140 em uma variedade de modos. Por exemplo, se a taxa de erro estiver abaixo de um limite, o compilador 130 pode determinar a compilação e envio somente de metade das codificações disponíveis para a dada unidade de dados. Inversamente, se a taxa de erros estiver em ou acima do limite, o compilador 130 pode determinar a compilação e envio de todas as codificações disponíveis para a data unidade de dados.

O dispositivo de armazenagem/receptor 160 pode ser, por exemplo, qualquer dispositivo capaz de receber as codificações compiladas enviadas pelo compilador 130. Por exemplo, o dispositivo de armazenagem/receptor 160 pode incluir um ou mais de vários dispositivos de armazenagem comumente disponíveis, incluindo, por exemplo, um disco rígido, um disco de servidor, ou um dispositivo de armazenagem portátil. Em várias implementações as codificações compiladas são enviadas diretamente para armazenagem após compilação, para exibição posterior ou transmissão adicional. O dispositivo de armazenagem/receptor 160 também pode ser, por exemplo, um dispositivo de computação capaz de receber dados codificados e processar os dados codificados. Tais dispositivos de computação podem incluir, por exemplo, set-top boxes, codificadores, decodificadores ou codecs.Tais dispositivos de computação também podem fazer parte de ou incluir, por exemplo, um dispositivo de exibição de vídeo como uma televisão. Um tal receptor pode ser projetado para receber dados transmitidos de acordo com um padrão específico.

A figura 2 mostra um diagrama de blocos de um sistema 200 para enviar e receber

dados codificados. O sistema 200 corresponde a uma implementação específica do sistema 100. O sistema 200 inclui duas fontes possíveis de codificações que são um codificador 210a e uma memória 210b. Essas duas fontes são conectadas a um compilador 230 através de um percurso 220, e o compilador 230 é adicionalmente conectado através de um percurso 250 a um receptor 260. Os percursos 220 e 250 são análogos aos percursos 120 e 150.

O codificador 210a recebe uma sequência de vídeo de entrada e inclui um codificador primário 212 e um codificador redundante 214. O codificador primário 212 cria codificações primários para cada uma das imagens (ou outras unidades de dados) na sequência de vídeo de entrada, e o codificador redundante 214 cria uma ou mais codificações redundantes para cada uma das imagens na sequência de vídeo de entrada. Observe que uma imagem pode incluir, por exemplo, um campo ou um quadro. O codificador 210a também inclui um multiplexor 216 que recebe, e multiplexa tanto a codificação primária como uma ou mais codificações redundantes para cada imagem. O multiplexor 216 cria, desse modo, um fluxo multiplexado, ou sinal, de codificações para a sequência de vídeo de entrada. O fluxo multiplexado é fornecido para qualquer um ou ambos entre a memória 210b ou compilador 230.

O compilador 230 recebe um fluxo de codificações a partir de qualquer um ou ambos o codificador 210a ou a memória 210b. O compilador 230 inclui um analisador 232, um seletor 234, um duplicador 236, e um multiplexor 238 conectados em série. O analisador 232 também é conectado a uma unidade de controle 231, e conectado diretamente ao multiplexor 238. Além disso, o seletor 234 tem duas conexões com o duplicador 236, incluindo uma conexão de fluxo 234a e uma conexão de controle 234b. Análogo ao compilador 130, o compilador 230 seleciona pelo menos algumas das múltiplas codificações para enviar para o receptor 260 e compila as codificações selecionadas para enviar as codificações selecionadas para o receptor 260. Além disso, o compilador 230 baseia a seleção, pelo menos em parte, em informações recebidas a partir do receptor 260.

A unidade de controle 231 recebe uma solicitação para enviar uma ou mais codificações. Uma tal solicitação pode vir de, por exemplo, o codificador 210a, o receptor 260, ou um dispositivo não mostrado no sistema 200. Um tal dispositivo pode incluir, por exemplo, um servidor de rede como anteriormente descrito. Por exemplo, a unidade de controle 231 pode receber uma solicitação do codificador 210a através do percurso 220 ou pode receber uma solicitação do receptor 260 através do percurso 250 ou pode receber uma solicitação autogerada a partir de um evento temporizado como anteriormente descrito. Após recebimento da solicitação, a unidade de controle 231 passa a solicitação para o analisador 232 e o analisador 232 solicita o fluxo correspondente de codificações a partir do codificador 210a ou memória 210b.

Implementações do sistema 200 não necessitam fornecer uma solicitação ou usar a unidade de controle 231. Por exemplo, o analisador 232 pode simplesmente compilar e en-

via codificações após recebimento de um fluxo de codificações a partir do codificador 210a.

O analisador 232 recebe o fluxo a partir do codificador 210a e separa o fluxo recebido em um sub-fluxo para as codificações primárias e um sub-fluxo para as codificações redundantes. As codificações redundantes são fornecidas ao seletor 234.

O seletor 234 também recebe informações a partir do receptor 260 que descrevem as condições atuais do percurso 250. Com base nas informações recebidas a partir do receptor 260, o seletor 234 determina qual das codificações redundantes incluir no fluxo de codificações que serão enviadas para o receptor 260. As codificações redundantes selecionadas são transmitidas a partir do seletor 234 na conexão de fluxo 2345a para o duplicador 236 e as codificações redundantes não selecionadas não são enviadas.

Em uma implementação, o seletor 234 recebe a partir do receptor 260 informações indicando a capacidade disponível no percurso 250, e o seletor 234 selecionada todas as codificações redundantes até que a capacidade esteja cheia. Por exemplo, as informações podem indicar que o percurso 250 tem uma capacidade de 2 Mbps (megabits/segundo) no momento atual. A capacidade pode ser variável, por exemplo, devido ao uso variável por outros compiladores (não mostrados). Considerando, por exemplo, que o compilador 230 dedique 1 Mbps às codificações primárias, o seletor 234 pode então dedicar o 1 Mbps restante para as codificações redundantes. Além disso, o seletor 234 pode então selecionar codificações redundantes até que largura de banda de 1 Mbps esteja cheia. Por exemplo, para encher a largura de banda de 1 Mpbs, o seletor 234 pode alocar para as codificações redundantes quatro partições em um esquema de acesso múltiplo por divisão de tempo no qual cada partição recebe 250 kbps.

O seletor 234 também pode selecionar uma codificação redundante dada duas vezes. Por exemplo, suponha que uma imagem dada foi alocada 1 Mbps para codificações redundantes, e a imagem específica tenha somente duas codificações redundantes que têm uma exigência de largura de banda de 1.200 kbps e 500 bps. O seletor 234 pode determinar que a segunda codificação redundante deve ser enviada duas vezes de modo a utilizar o 1 Mbps inteiro. Para obter isso, o seletor 234 envia a segunda codificação redundante no fluxo através da conexão de fluxo 234a para o duplicador 236, e também envia um sinal de controle para o duplicador 236 através da conexão de controle 234b. O sinal de controle instrui o duplicador 236 a duplicar a segunda codificação redundante e incluir a codificação duplicada no fluxo que o duplicador 236 envia para o multiplexor 238.

O receptor 260 inclui um receptor de dados 262 conectado a uma fonte de informações de canal 264. O receptor de dados 262 recebe o fluxo de codificações enviadas do compilador 230 através do percurso 250, e o receptor de dados 262 pode executar uma variedade de funções. Tais funções podem incluir, por exemplo, decodificar as codificações, e exibir a sequência de vídeo decodificada. Outra função do receptor de dados 262 é determi-

nar informações de canal para fornecer para a fonte de informações de canal 264. As informações de canal fornecidas à fonte de informação de canal 264, a partir do receptor de dados 262, indicam condições atuais do percurso 250. Essas informações podem incluir, por exemplo, uma taxa de erro como uma taxa de erro de bit ou uma taxa de erro de pacote, ou utilização de capacidade como a taxa de dados que está sendo utilizada ou a taxa de dados que ainda está disponível. A fonte de informação de canal 264 provê essas informações ao seletor 234 como anteriormente descrito. A fonte de informações de canal 264 pode fornecer essas informações através do percurso 250 ou através de outro percurso, como por exemplo, um canal de apoio ou um canal auxiliar.

O sistema 200 não é específico a nenhum algoritmo de codificação particular, muito menos a um padrão inteiro. Entretanto, o sistema 200 pode ser adaptado ao padrão H.264/AVC. Em uma tal implementação, o codificador 210a é adaptado para operar como um codificador H.264/AVC por exemplo, por adaptar o codificador primário 212 para criar PCPs e adaptar o codificador redundante 214 para criar RCPs. Além disso, nessa implementação o analisador 232 é adaptado para analisar as PCPs em um sub-fluxo enviado diretamente para o multiplexor 238, e analisar as RCPs em um sub-fluxo enviado ao seletor 234. Adicionalmente, nessa implementação o receptor 260 é adaptado para operar como um decodificador H.264/AVC, além de fornecer as informações de canal.

As figuras 3 e 4 apresentam fluxogramas de processos para utilizar os sistemas 100 e 200. Esses fluxogramas serão descritos resumidamente e então vários aspectos serão explicados em maior detalhe em combinação com figuras adicionais.

A figura 3 mostra um fluxograma que descreve um processo 300 que pode ser executado por cada um dos sistemas 100 e 200. O processo 300 inclui receber uma solicitação para enviar através de um canal uma ou mais codificações de pelo menos uma porção de um objeto de dados (305). Por exemplo, no sistema 100 o compilador 130 pode receber uma solicitação para enviar codificações para o dispositivo de armazenagem/receptor 160. Como outro exemplo, no sistema 200 a unidade de controle 231 do compilador 230 pode receber uma solicitação para enviar codificações para o receptor 260.

O processo 300 inclui ainda acessar informações para determinar, ou selecionar, qual das múltiplas codificações de pelo menos uma porção de um objeto de dados enviar através de um canal (310). As informações são tipicamente acessadas após receber a solicitação em operação 305 e as informações podem ser acessadas em resposta ao recebimento da solicitação. Por exemplo, no sistema 100 o compilador 130 acessa informações fornecidas pela fonte de informações 140 e no sistema 200 o seletor 234 acessa informações de canal fornecidas pela fonte de informações de canal 264.

O processo 300 inclui ainda determinar, com base nas informações acessadas, um conjunto das múltiplas codificações a enviar através do canal (320). O conjunto é determi-

nado a partir de múltiplas codificações, inclui pelo menos uma e possivelmente mais de uma das múltiplas codificações. Além disso, o número de codificações no conjunto se baseia nas informações acessadas. Por exemplo, no sistema 100 o compilador 130 seleciona qual das codificações enviar através do percurso 150 e a quantidade de codificações selecionadas depende das informações acessadas. Como outro exemplo, no sistema 200 o seletor 234 seleciona qual das codificações redundantes enviar através do percurso 250, e a quantidade selecionada depende das informações de canal acessadas. Em muitas implementações, a quantidade será de pelo menos duas. Entretanto, a quantidade pode ser zero ou uma em outras implementações.

O compilador 130 pode incluir, por exemplo, um servidor de dados, um computador pessoal, um servidor de rede, um servidor de vídeo ou um codificador de vídeo. Em muitas implementações, porções diferentes do compilador 130 executam as diferentes operações do processo 300, com as diferentes porções incluindo as instruções de hardware e software necessárias para executar a operação específica. Desse modo, por exemplo, uma primeira porção de um servidor de vídeo pode receber a solicitação (305), uma segunda porção do servidor de vídeo pode acessar as informações (310), e uma terceira porção do servidor de dados pode determinar o conjunto de codificações a enviar (320).

A figura 4 mostra um fluxograma que descreve um processo 400 que pode ser executado por cada um dos sistemas 100 e 200. O processo 400 inclui fornecer informações para determinar qual de múltiplas codificações de pelo menos uma porção de um objeto de dados enviar através de um canal (410). Por exemplo, no sistema 100 a fonte de informações 140 provê tais informações para o compilador 130, e no sistema 200 a fonte de informações de canal 264 provê tais informações para o seletor 234.

O processo 400 inclui ainda receber através do canal um conjunto de codificações pelo menos da porção do objeto de dados (420). O conjunto de codificações inclui pelo menos uma e possivelmente mais de uma das múltiplas codificações. Além disso, o número de codificações no conjunto se baseia nas informações fornecidas em operação 410. Por exemplo, no sistema 100 o dispositivo de armazenagem/receptor 160 recebe através do percurso 150 um conjunto de codificações determinado e enviado pelo compilador 130. Além disso, o compilador 130 seleciona as codificações no conjunto com base nas informações recebidas a partir da fonte de informações 140. Como outro exemplo, no sistema 200 o receptor 260 recebe através do percurso 250 uma quantidade de codificações em um conjunto selecionado e enviado pelo compilador 230. Além disso, o compilador 230 determina as codificações a incluir no conjunto com base nas informações recebidas a partir da fonte de informações de canal 264.

As figuras 5 e 8 apresentam processos adicionais para utilizar o sistema 200. As figuras 6-7 apresentam diagramas que serão explicados em combinação com as figuras 5 e

8.

A figura 5 mostra um fluxograma que descreve um processo 500 que pode ser executado pelo sistema 200. O processo 500 inclui codificar múltiplas codificações para cada imagem em uma unidade de dados, como, por exemplo, um grupo de imagens ("GOP") ou apenas uma única imagem (510). No sistema 200, o codificador 210a cria múltiplas codificações para cada imagem em uma unidade de dados utilizando o codificador primário 212 e o codificador redundante 214. Um exemplo de múltiplas codificações é mostrada na figura 6.

A figura 6 inclui uma representação pictorial 600 de múltiplas codificações para cada uma de N imagens. As codificações podem ser criadas de acordo com o padrão H.264/AVC para produzir as PCPs e RCPs. Para cada imagem, uma PCP e múltiplas RCPs são mostradas. Especificamente, as PCPs mostradas incluem uma PCP 1 (605), uma PCP 2 (610) e uma PCP N (615). Além disso, as RCPs mostradas incluem (1) uma RCP 1.1 (620) e uma RCP 1.2 (625), correspondendo à PCP 1 (605), (2) uma RCP 2.1 (630), uma RCP 2.2 (635), uma RCP 2.3 (640) e uma RCP 2.4 (645), correspondendo à PCP 2 (610) e (3) uma RCP N.1 (650), uma RCP N.2 (655) e uma RCP N.3 (660), correspondendo a PCP N 615). As imagens codificadas podem ser criadas utilizando uma ou mais de uma variedade de técnicas de codificação.

As múltiplas codificações mostradas na representação 600, bem como as codificações em muitas outras implementações, são codificações de fonte. codificações de fonte são codificações que comprimem os dados sendo codificados, como comparado com codificações de canal que são codificações que acrescentam informações adicionais que são utilizadas tipicamente para correção ou detecção de erros. Desse modo, em implementações nas quais múltiplas codificações de fonte são enviadas para uma dada imagem, as múltiplas codificações de fonte fornecem redundância de codificação de fonte. Redundância é valiosa, por exemplo, quando canais de perda são utilizados como ocorre com muitas das implementações de transmissão de vídeo discutidas aqui.

O processo 500 inclui ainda armazenar as múltiplas codificações (520). As codificações podem ser armazenadas, por exemplo, em qualquer de uma variedade de dispositivos de armazenagem. Como com muitas das operações no processo 500, e os outros processos revelados nesse pedido, a operação 520 é opcional. A armazenagem é opcional no processo 500 porque, por exemplo, em outras implementações as múltiplas codificações são processadas,por exemplo, por um compilador diretamente após serem criadas. No sistema 300, as múltiplas codificações podem ser armazenadas na memória 210b.

O processo 500 inclui receber uma solicitação para enviar codificações da imagem ou imagens, na unidade de dados (530). No sistema 200, uma solicitação para enviar codificações pode ser recebida pela unidade de controle 231 como anteriormente descrito.

O processo 500 inclui acessar informações de canal para determinar qual das múl-

tiplas codificações preparadas da imagem, ou imagens, na unidade de dados enviar através do percurso 250 (540). O processo 500 inclui ainda determinar um conjunto de codificações a enviar através do percurso 250, com o conjunto determinado incluindo pelo menos uma e possivelmente mais das múltiplas codificações, e o número de codificações no conjunto sendo baseado nas informações de canal acessadas (550). As operações 540 e 550 são análogas às operações 310 e 320 no processo 300, e o desempenho de operações 310 e 320 pelo sistema 200 foi explicado, por exemplo, na discussão acima de operações 310 e 320. Uma explicação adicional será fornecida, entretanto, utilizando a figura 7.

A figura 7 inclui uma representação pictorial 700 das codificações selecionadas para cada de N imagens. As codificações selecionadas foram selecionadas a partir das codificações mostradas na representação 600. Como mostrado na representação 700, todas as PCPs são selecionadas. Isto é, a PCP 1 (605), a PCP 2 (610) e a PCP N (615). São selecionadas. Entretanto, nem todas as RCPs disponíveis na representação 600 são selecionadas. Especificamente, (1) para a PCP 1 (605), a RCP 1.1 (620) é selecionada, porém a RCP 1.2 (625) não é selecionada, (2) para a PCP 2 (610), a RCP 2.1 (630) e a RCP 2.2 (635) são selecionadas, porém a RCP 2 3 (640) e a RCP 2.4 (645) não são selecionadas, e (3) para a PCP N (615), a RCP N.1 (650) e a RCP N.2 (655) são selecionadas, porém a RCP N.3 (660) não é selecionada. Adicionalmente, a RCP 2.'1 (630) é selecionada duas vezes, de modo que a RCP 2.1 (640) aparecerá duas vezes em um fluxo de codificações multiplexado, final. As duas seleções da RCP 2.1 (630) são designadas com numerais de referência 730a e 730b na representação 700.

A figura 7 mostra o resultado do processo de seleção para um exemplo, porém a figura 7 não descreve porque algumas codificações foram selecionadas e outras não foram. Vários critérios podem ser utilizados para determinar qual das possíveis codificações selecionar. Por exemplo, as codificações podem ser selecionadas na ordem recebida para uma dada imagem até que uma restrição de bit para aquela imagem seja utilizada. Como outro exemplo, um valor de uma métrica de distorção pode ser calculado para cada codificação e todas as codificações tendo um valor de distorção abaixo de um limite específico podem ser selecionadas. O apêndice A descreve o processo de seleção para outra implementação.

O processo 500 inclui ainda enviar as codificações selecionadas (560). Como descrito anteriormente, as codificações podem ser enviadas, por exemplo, para um dispositivo de armazenagem ou um dispositivo de processamento. No sistema 200, o compilador 230 envia o fluxo multiplexado de codificações a partir do multiplexor 238 através do percurso 250 para o receptor 260. Muitas implementações enviam as codificações pela formação de um fluxo que inclui as codificações selecionadas.

Deve ficar claro que a quantidade de redundância de codificação de fonte que é fornecida pode variar para imagens diferentes. A quantidade de redundância de codificação

de fonte pode variar devido, por exemplo, a diferentes números de codificações de fonte sendo selecionadas. Números diferentes de codificações de fonte podem ser selecionadas para diferentes imagens porque, por exemplo, as codificações de fonte para diferentes imagens tendo tamanhos diferentes ou as informações acessadas sendo diferentes para imagens diferentes.

A figura 8 mostra um fluxograma que descreve um processo 800 que pode ser executado pelo receptor 260 do sistema 200. O processo 800 inclui determinar informações de canal para uso na determinação de qual das múltiplas codificações enviar através de um canal (810) e então fornecer essas informações (820). No sistema 200, o receptor de dados 262 determina informações de canal indicando condições atuais do canal e provê essas informações de canal para a fonte de informações de canal 264. A fonte de informações de canal 264 provê então as informações de canal para o seletor 234. A operação 820 do processo 800 é análoga à operação 410 do processo 400.

O processo 800 inclui ainda receber através do canal um conjunto, ou uma quantidade, de codificações (830). O conjunto inclui uma, e possivelmente mais de uma, das múltiplas codificações. A quantidade de codificações no conjunto tendo sido selecionada com base nas informações de canal fornecidas e então enviada através do canal. A operação 830 do processo 800 é análoga à operação 420 do processo 400, e um exemplo do sistema 200 que executa operação 420 foi fornecido acima na discussão da operação 420. O processo 800 inclui ainda processar as codificações recebidas (840). Os exemplos de processamento incluem decodificar as codificações, exibir as codificações decodificadas, e enviar as codificações recebidas ou as codificações decodificadas para outro destino.

Em uma implementação, o sistema 200 adere ao padrão H.264/AVC. O padrão H.264/AVC define uma variável denominada "redundant_pic_count" que é zero para uma PCP e é não zero para uma RCP. Além disso, a variável é incrementada para cada RCP que é associada a uma PCP dada. Desse modo, o receptor 260 é capaz de determinar, para cada imagem, se qualquer codificação recebida específica é uma RCP ou uma PCP. Para cada imagem, o receptor 260 pode então decodificar e exibir a codificação com o valor mais baixo para a variável "redundant_pic_count". Entretanto, outras implementações podem combinar múltiplas imagens codificadas que são recebidas sem erro.

As figuras 9-11 referem-se a outra implementação que organiza codificações em camadas e provê resiliência de erro. A figura 9 mostra um diagrama de blocos de um sistema 900 que inclui um codificador 910a que provê codificações para um compilador 930, e o compilador 930 provê codificações compiladas para um receptor 960. O sistema 900 inclui ainda a fonte de informações 140. A estrutura e operação do sistema 900 é amplamente análoga àquela do sistema 200, com numerais de referência correspondentes genericamente tendo pelo menos algumas funções correspondentes. Por conseguinte, características

idênticas não serão necessariamente repetidas, e a discussão do sistema 900 que se segue focaliza nas diferenças a partir do sistema 200.

O codificador 910a inclui o codificador primário 212 e o codificador redundante 214. O codificador 910a inclui ainda um gerador de distorção 915 que recebe codificações a partir do codificador redundante 214, gera um valor de uma métrica de distorção para cada codificação, e provê cada codificação e o valor de distorção para cada codificação para uma unidade de ordenação 917. A unidade de ordenação 917 ordena as codificações baseadas nos valores de distorção gerados, e provê as codificações ordenadas para um multiplexor 916. O multiplexor 916 é análogo ao multiplexor 216 e multiplexa as codificações redundantes ordenadas e as codificações primárias em um fluxo de saída que é fornecido ao compilador 930.

O compilador 930 inclui a unidade de controle 231 conectada a um analisador 932 que provê entrada tanto para uma unidade de camada 937 como um multiplexor 938. A unidade de camada 937 também provê entrada para o multiplexor 938. O compilador 930 recebe o fluxo de codificações a partir do codificador 910a e provê um fluxo compilado de codificações para o receptor 960.

O analisador 932 é análogo ao analisador 232, e separa o fluxo recebido em codificações primárias que são fornecidas diretamente ao multiplexor 938 e codificações secundárias que são fornecidas à unidade de camadas 937. Mais especificamente, o analisador 932 separa o fluxo recebido em uma camada de base para as codificações primárias e um sub-fluxo para as codificações redundantes. O analisador 932 provê a camada de base para o multiplexor 938, e provê o sub-fluxo de codificações redundantes para a unidade de camadas 937.

O sub-fluxo de codificações redundantes que a unidade de camadas 937 recebe inclui codificações redundantes que foram ordenadas pela unidade de ordenação 917. A unidade de camadas 937 separa o sub-fluxo de codificações redundantes em uma ou mais camadas, mencionadas como camadas de intensificação, e provê as camadas de intensificação para o multiplexor 938 como necessário. Como mostrado na figura 9, a unidade de camada 937 tem "n" saídas 937a-937n, uma para cada camada de intensificação. Se uma implementação exigir somente uma camada de intensificação, então a unidade de camada 937 necessitaria somente de uma saída para camadas de intensificação, e forneceria a única camada de intensificação na saída 937a. Os sistemas podem incluir múltiplas saídas 937a-n, entretanto, fornecendo flexibilidade para várias implementações que podem exigir números diferentes de camadas.

A unidade de camadas 937 recebe também entrada a partir da fonte de informações 140 e utiliza essas informações em um modo análogo àquele descrito para uso pelo compilador 130 das informações a partir da fonte de informações 140, bem como uso pelo

seletor 234 das informações de canal provenientes da fonte de informações de canal 264. Em particular, a unidade de camadas 937 pode utilizar as informações provenientes da fonte de informações 140 para determinar quantas camadas de intensificação deve criar.

Várias implementações do compilador 930 operam em conjuntos discretos de imagens. Por exemplo, uma implementação de vídeo opera em um GOP. Nessa implementação, o analisador 932 provê uma camada de base separada para o multiplexor 938 para cada GOP, e a unidade de camadas 937 provê camadas de intensificação separadas para cada GOP.

O receptor 960 é genericamente análogo ao receptor 160, e inclui um receptor de dados 962 que recebe o fluxo multiplexado proveniente do multiplexor 938. O receptor de dados 962 é análogo ao receptor 262, e pode executar uma variedade de funções. Tais funções podem incluir, por exemplo, decodificar as codificações, e exibir ou de outro modo fornecer as codificações decodificadas para um usuário final.

O sistema 900 não é específico a nenhum algoritmo de codificação particular, muito menos a um padrão inteiro. Entretanto, o sistema 900 pode ser adaptado ao padrão H.264/AVC. Em uma tal implementação, o codificador 910a é adaptado para operar com um codificador H.264/AVC por exemplo, por adaptar o codificador primário 212 para criar PCPs e adaptar o codificador redundante 214 para criar RCP's. Além disso, o analisador 932 é adaptado para analisar as PCPs em um sub-fluxo enviado diretamente para o multiplexor 938, e para analisar as RCPs em um sub-fluxo enviado à unidade de camadas 937. Adicionalmente, o receptor 960 é adaptado para operar como um decodificador H.264/AVC.

A figura 10 provê um fluxograma de uma implementação de um processo 1000 para operar o sistema 900 em um ambiente de vídeo. O processo 1000 inclui codificar múltiplas codificações, incluindo uma codificação primária e uma ou mais codificações redundantes, para cada imagem em uma sequência de vídeo (1010). A operação 1010 é análoga à operação 510 no processo 500. Na figura 9, o codificador primário 212 e o codificador redundante 214 pode criar as codificações para operação 1010. Em uma implementação, as codificações criadas podem incluir as codificações mostradas na representação pictorial 600.

O processo 1000 inclui gerar, ou de outro modo determinar, um valor de uma métrica de distorção para cada uma das codificações redundantes (1020). A métrica de distorção pode ser qualquer métrica, ou medição, por exemplo, para classificar as codificações de acordo com alguma medição de qualidade. Uma tal medição, determinada para cada codificação dada, é o erro médio quadrado ("MSE") entre a codificação dada e a imagem original. Outra tal medição é a relação de sinal para ruído de pico ("PSRN") para a codificação dada. Em muitas implementações, o MSE é calculado entre uma imagem decodificada e a imagem original, e tipicamente mediada através de um grupo de imagens para produzir uma métrica mencionada como o MSE médio. Em muitas implementações, a PSNR para uma codifica-

ção é calculada a partir do MSE como uma função logarítmica do MSE para aquela codificação, como é bem sabido. O conjunto de PSNRs para um conjunto de codificações pode ser medido por somar e dividir, como é bem conhecido; para produzir a PSNR média. Entretanto, a PSNR média pode ser alternativamente calculada diretamente a partir do MSE médio utilizando a mesma função logarítmica para calcular PSNR para uma codificação individual. A computação alternativa da PSNR média põe mais peso sobre imagens decodificadas que têm distorção grande, e esse peso tende a refletir mais precisamente a variação de qualidade percebida por um usuário final vendo as imagens decodificadas. Outras métricas de distorção também podem ser utilizadas.

O processo 1000 inclui ordenar as codificações com base no valor de distorção gerado para cada codificação (1030), e organizar as codificações ordenadas em camadas (1035). A unidade de ordenação 917 pode executar tanto a ordenação como a formação em camadas. Em uma implementação, a ordenação ocorre por reorganizar as codificações redundantes de modo que estejam em ordem crescente de valor de distorção (espera-se que valores de distorção mais elevados resultem em decodificações que são de qualidade mais desfavorável). A reorganização pode ser, por exemplo, reorganização física ou reorganização lógica. Reorganização lógica inclui, por exemplo, criar uma lista com link a partir das codificações, com cada codificação em uma camada apontando para a seguinte codificação em sua camada.

Além disso, a formação em camadas pode ocorrer por atribuir um certo número de bits a cada camada de codificações redundantes, e então encher as camadas com as codificações ordenadas de tal modo que cada camada esteja cheia antes de se mover para encher uma camada sucessiva. Em outra implementação, a formação de camadas pode ocorrer por dividir o fluxo de codificações em camadas com base nos valores da métrica de distorção. Por exemplo, todas as codificações redundantes com um valor de distorção entre certos pontos finais podem ser colocadas em uma camada comum.

A figura 11 provê uma representação pictorial 1100 das codificações provenientes da representação 600 após as codificações terem sido ordenadas em múltiplas camadas de acordo com uma implementação do processo 1000. Especificamente, a representação 110 mostra que as codificações foram organizadas em quatro camadas, incluindo uma Camada de base 1110, uma Camada de intensificação 1 1120, uma Camada de Intensificação 2 1130 e uma Camada de intensificação 3 1140.

A Camada de base 1110 inclui todas as PCPs para um dado GOP. As PCPs mostradas são a PCP 1 (605), a PCP 2 (610), e a PCP N (615). A camada de intensificação 1 1120 é a primeira camada de codificações redundantes e inclui a RCP 1.1 (620), a RCP 2.1 (630), a RCP 2.2 (635) e a RCP N.1 (650). A Camada de intensificação 2 1130 é a segunda camada de codificações redundantes e inclui a RCP 2.3 (640), a RCP 2.4 (645) e a RCP N.2

(655). A Camada de intensificação 3 1140 é a terceira camada de codificações redundantes e inclui a RCP 1.2 (625) e a RCP N.3 (660). Nessa implementação, as Camadas de Intensificação são organizadas em ordem de valores de distorção crescentes, de tal modo que as "melhores" codificações redundantes são incluídas nas Camadas de Intensificação iniciais.

Com referência novamente ao Apêndice A, é mostrada uma implementação para selecionar codificações com base em valores de distorção. A implementação pode ser estendida para ordenar um conjunto de codificações através de um GOP inteiro, por exemplo, em vez de simplesmente ordenar um conjunto de codificações para uma imagem dada. Em uma tal extensão, os valores esperados da redução de distorção são determinados com relação ao GOP inteiro em vez de uma única imagem, e os valores esperados de redução de distorção são otimizados através de todas as codificações para o GOP em vez apenas das codificações para a imagem única. Observa-se também que no Apêndice A os valores esperados de distorção para uma sequência se baseiam nos valores de distorção calculados para codificações individuais na sequência.

O processo 1000 inclui armazenar as codificações e os valores de distorção (1040). Essa operação, como com muitas no processo 1000, é opcional. A operação 1040 é análoga à operação 520 no processo 500. Implementações podem, por exemplo, recuperar codificações anteriormente armazenadas. Inversamente, implementações podem receber codificações atualmente geradas.

O processo 1000 inclui receber uma solicitação para enviar uma ou mais codificações para uma imagem dada (1050). O processo 1000 inclui ainda acessar informações para determinar as codificações a enviar para a imagem dada (1060), determinar a última codificada a enviar com base nas informações acessadas (1070), e enviar as codificações selecionadas (1080). As operações 1050, 1060, 1070 e 1080 são análogas às operações 530-560 no processo 500, respectivamente.

Em uma implementação, as informações acessadas na operação 1060 a partir da fonte de informações 140 são utilizadas para determinar quantos bits podem ser utilizados para enviar as codificações de uma imagem dada. Como as codificações redundantes já são ordenadas por seus valores de distorção, a ordem representa presumivelmente a preferência para quais codificações redundantes selecionar e enviar. Por conseguinte, para a imagem dada, a codificação primária é selecionada e incluída no conjunto de codificações a enviar, e todas as codificações redundantes são selecionadas, em ordem, até que o número disponível de bits tenha sido utilizado. Pode ocorrer que, para uma imagem dada, haja alguns bits restantes que não são utilizados pelas codificações selecionadas, porém que esses bits restantes não são suficientes para enviar a codificação seguinte no conjunto ordenado de codificações para a imagem dada. Um método de resolver um tal cenário é arredondar para cima ou para baixo, decidindo efetivamente fornecer os bits extra para as codi-

ficações da imagem seguinte ou retirar alguns bits das codificações da imagem seguinte. Por conseguinte, nessa implementação, as codificações são selecionadas por determinar quantos bits são disponíveis e então terminar o fluxo de codificações ordenadas no valor de bit determinado (talvez arredondar para cima ou para baixo). Desse modo, uma quantidade de codificações é selecionada por selecionar a codificação na qual terminar o fluxo. Isto é, uma quantidade de codificações é selecionada por selecionar uma "última" codificação a enviar. As codificações selecionadas são incluídas no conjunto de codificações a enviar.

Na implementação acima, a operação (1070) de determinar a última codificação a enviar também pode ser executada simplesmente selecionando quantas camadas enviar. Por exemplo, se a implementação já determinou o número de bits para enviar as codificações de uma imagem dada, o processo pode terminar o fluxo de codificações ordenadas ao término da camada na qual o valor de bit determinado está compreendido. Desse modo, se a Camada de base e cada Cama de Intensificação exigir 1000 bits, e as informações acessadas a partir da fonte de informações 140 indicar que 2700 bits são disponíveis, então uma implementação seleciona a Camada de Base e as primeiras duas Camadas de Intensificação a enviar. Como 3000 bits seriam utilizados, essa implementação também pode subtrair 300 bits da divisão de bits da imagem seguinte.

Na implementação acima, como com muitas implementações, múltiplas fatias podem ser utilizadas para codificar uma RCP dada. Em uma implementação típica, todas essas fatias serão colocadas na mesma camada para assegurar que todas (ou nenhuma) as fatias para aquela RCP são enviadas. Entretanto, em alguma implementação todas as fatias para uma RCP data não são colocadas na mesma camada.

Em outra implementação, as codificações são organizadas em camadas (1035) somente após selecionar quais codificações enviar (1070). Por exemplo, no sistema 900, a unidade de camada 937 pode organizar as codificações em camadas. Essa implementação pode oferecer vantagens de flexibilidade porque as informações acessadas a partir da fonte de informações 140 podem ser utilizadas na determinação dos tamanhos de camadas. Adicionalmente, a unidade de camada 937 pode gerar também os valores de distorção para as codificações e executar a ordenação. Por gerar os valores de distorção na unidade de camada 937, a unidade de camada 937 pode ter a vantagem de já ter acessado informações a partir da fonte de informações 140. As informações acessadas podem permitir, por exemplo, que a unidade de camadas 937 gere valores de distorção que levem em consideração o número disponível de bits (ou camadas ou codificações) que pode ser enviado.

Implementações do processo 1000 também podem fornecer redimensionamento de resiliência de erro ao fluxo de codificações. Para fornecer redimensionamento de resiliência de erro, é desejável que o fluxo tenha um aumento incremental em resiliência de erro à medida que o número de codificações no fluxo é aumentado. Isto é, o valor esperado de uma

medição de erro (ou distorção, por exemplo) diminui à medida que mais codificações são enviadas. Considerando ambientes de vídeo, e implementações de H.264/AVC em particular, uma implementação redimensionável resiliente em erro específico envia primeiramente as PCPs para as imagens de um GOP, e então envia as RCPs. À medida que mais codificações são enviadas, iniciando com as PCPs e continuando com as RCPs, a resiliência de erro do GOP é aumentada porque a implementação tem uma probabilidade mais elevada de decodificar corretamente o GOP. Adicionalmente, se as codificações tiverem sido ordenadas de acordo com valores de distorção crescentes, então a série de codificações que é selecionada para qualquer imagem dada pode ser ótima, ou próxima à ótima, para a taxa de bits sendo utilizada. Deve ser claro que redimensionamento de resiliência de erro pode ser fornecida com ou sem camadas.

O sistema 900 pode ser também modificado de tal modo que várias operações são opcionais por selecionar um modo. Por exemplo, um usuário pode indicar que valores de distorção não são necessários, e o sistema pode desabilitar o gerador de distorção 915 e a unidade de ordenação 917. O usuário pode indicar também que a formação de camadas não é necessária, e o sistema pode fazer com que a unidade de camada 937 opere como o seletor 234 e duplicador 236. Além disso, deve ser evidente que em uma implementação, um sistema pode ser induzido a operar como, por exemplo, o sistema 200 ou o sistema 900, com o uso, por exemplo, de comutadores para habilitar ou desabilitar vários recursos que são específicos para o sistema 200 ou sistema 900.

Também deve ser claro que as funções do duplicador 236 podem ser implementadas na unidade de camadas 937, por exemplo, de tal modo que as codificações específicas possam ser duplicadas e incluídas em uma camada. Por exemplo, se houver bits não utilizados após a seleção de uma camada, então a última camada pode ser estendida por duplicar uma ou mais codificações.

Muitas implementações estão em conformidade com o padrão H.264/AVC, embora todas as implementações não necessitem estar em conformidade com o padrão H.264/AVC ou qualquer outro padrão. Além disso, muitas implementações são descritas utilizando termos associados ao padrão H.264/AVC, como por exemplo, "imagem codificada primária", "imagem codificada redundante", e "fatia redundante". Entretanto, o uso de tais termos não quer dizer que a implementação é, ou necessita estar em conformidade com o padrão H.264/AVC. Esses termos são utilizados em um sentido geral, independente do padrão H.264/AVC, e não incorporam o padrão H.264/AVC ou qualquer outro padrão. Ainda adicionalmente, esses termos podem ser utilizados com outros padrões, incluindo padrões futuros, e as implementações pretendem ser aplicáveis com tais padrões.

Implementações também podem operar por acessar informações a partir da fonte de informações 140 e então criar as codificações desejadas em vez de selecionar entre co-

dificações preparadas. Essas implementações podem ter a vantagem, por exemplo, de ser mais flexíveis em atender restrições específicas de taxa de bits.

Como descrito anteriormente, muitas implementações determinam um conjunto de codificações a enviar, onde a determinação se baseia em informações acessadas. Em muitas implementações, determinar o conjunto, com a determinação sendo baseada em informações de acesso, será equivalente a selecionar a quantidade de codificações, com a quantidade sendo baseada em informações acessadas. Entretanto, implementações podem existir nas quais as duas características diferem. adicionalmente, em implementações que acessam informações para selecionar qual de múltiplas codificações enviar através de um canal, as informações podem ser acessadas em resposta a uma solicitação para enviar através do canal uma ou mais codificações pelo menos da porção do objeto de dados.

Implementações podem otimizar em uma variedade de fatores diferentes em lugar de, ou além de, distorção. Tais outros fatores incluem, por exemplo, o custo de enviar dados através de um canal dado em uma dada qualidade.

Percursos (por exemplo, o percurso 110) podem ser diretos se o percurso não tiver elementos intermediários, ou indiretos que permitem elementos intermediários. Se dois elementos forem ditos como sendo "acoplados", os dois elementos podem ser acoplados, ou conectados, direta ou indiretamente. Além disso, um acoplamento necessita não ser físico, como, por exemplo, quando dois elementos são acoplados de forma comunicativa através do espaço livre através de vários roteadores e repetidores (por exemplo, dois telefones celulares).

Implementações dos vários processos e características descritos aqui podem ser incorporadas em uma variedade de equipamentos ou aplicações diferentes, particularmente, por exemplo, equipamentos ou aplicações associados à transmissão de vídeo. Os exemplos de equipamentos incluem codecs de vídeo, servidores de rede, telefones celulares, assistentes digitais portáteis ("PDAs"), set-top boxes, laptops e computadores pessoais. Como deve ser claro a partir desses exemplos, codificações podem ser enviadas através de uma variedade de percursos, incluindo, por exemplo, percursos cabeados ou sem fio, Internet, linhas de televisão a cabo, linhas telefônicas e conexões de Ethernet.

Os vários aspectos, implementações e características podem ser implementados em um ou mais de uma variedade de modos, mesmo se descritos acima sem referência a um modo específico ou utilizando somente um modo. Por exemplo, os vários aspectos, implementações e características podem ser implementados utilizando, por exemplo, um ou mais de (1) um método (também mencionado como processo), (2) um aparelho, (3) um aparelho ou dispositivo de processamento para executar um método, (4) um programa ou outro conjunto de instruções para executar um ou mais métodos, (5) um aparelho que inclui um programa ou um conjunto de instruções, e (6) um meio legível por computador.

Um aparelho pode incluir, por exemplo, hardware discreto ou integrado, firmware e software. Como exemplo, um aparelho pode incluir, por exemplo, um processador, que se refere a dispositivos de processamento em geral, incluindo, por exemplo, um microprocessador, um circuito integrado, ou um dispositivo lógico programável. Como outro exemplo, um aparelho pode incluir um ou mais meios legíveis por computador tendo instruções para realizar um ou mais processos.

Um meio legível por computador pode incluir, por exemplo, um portador de software ou outro dispositivo de armazenagem, como, por exemplo, um disco rígido, um disquete compacto, uma memória de acesso aleatória ("RAM"), ou uma memória somente de leitura ("ROM"). Um meio legível por computador também pode incluir por exemplo, ondas eletromagnéticas formatadas codificando ou transmitindo instruções. Instruções podem ser, por exemplo, em hardware, firmware, software ou em uma onda eletromagnética. Instruções podem ser encontradas, por exemplo, em um sistema operacional, uma aplicação separada ou uma combinação dos dois. Um processador pode ser caracterizado, portanto, como por exemplo, tanto um dispositivo configurado para realizar um processo como um dispositivo que inclui um meio legível por computador tendo instruções para realizar um processo.

Diversas implementações foram descritas. Não obstante, será entendido que várias modificações podem ser feitas. Por exemplo, elementos de implementações diferentes podem ser combinados, suplementados, modificados ou removidos para produzir outras implementações. Adicionalmente, uma pessoa com conhecimentos comuns entenderá que outras estruturas e processos podem ser substituídos por aqueles revelados e as implementações resultantes executarão pelo menos substancialmente a(s) mesma(s) função(ões), pelo menos substancialmente do(s) mesmo(s) modo(s), para obter pelo menos substancialmente o(s) mesmo(s) resultado(s) que as implementações reveladas. Por conseguinte, essas e outras implementações são consideradas por esse pedido e estão compreendidas no escopo das reivindicações a seguir.

APÊNDICE A

Uma implementação de seleção de fatias redundantes

Suponha que quando o fluxo de bits pré-codificado é gerado, múltiplas imagens redundantes são codificadas para cada imagem de entrada a fim de fornecer resiliência de erro diferente e equilíbrio de taxa de erro. Portanto, para uma dada taxa de perda de canal e restrição de taxa de bits, é possível selecionar um conjunto de fatias redundantes para incluir no fluxo de bits final a fim de maximizar sua capacidade de resiliência de erro.

A distorção do vídeo recebido pode ser dividida em duas partes: distorção de fonte devido à compressão e distorção de canal devido a perdas de fatia durante transmissão. Uma fatia redundante é somente usada quando sua fatia primária correspondente não é corretamente recebida. Portanto, fatias redundantes afetam somente a distorção de canal.

Suponha que uma sequência de vídeo de entrada tenha N imagens, e para a imagem n haja K_n fatias redundantes diferentes no fluxo de bits pré-codificado. Por incluir um conjunto S_n de fatias redundantes para a imagem n no fluxo de bits final, a distorção de canal esperada para a imagem pode ser reduzida e a quantidade de redução de distorção é indicada como $E[\Delta D_n]$. Observe que minimizar a distorção de canal para a imagem n é equivalente para maximizar $E[\Delta D_n]$.

Considere que para cada imagem n, $E[\Delta D_n]$.é aproximadamente não correlacionado. O objetivo da seleção de fatia redundante pode ser gravado como

$$\max \sum_{n=1}^{N} E[\Delta D_n] \quad \text{s.t.} \quad \sum_{n=1}^{N} R_n^{(RCP)} \le R_T - \sum_{n=1}^{N} R_n^{(PCP)} \qquad (1)$$

Na equação, R_T é a restrição de taxa dada,

$R_n^{(PCP)}$ e $R_n^{(RCP)}$ são as taxas para a fatia primária e as fatias redundantes para a imagem n, respectivamente. Além disso, para uma dada taxa de perda de fatia p, $E[\Delta D_n]$ pode ser expresso como

$$\begin{aligned} E[\Delta D_n] &= \sum_{i=1}^{\|S_n\|} E[\Delta D_n^i] \\ &= \sum_{i=1}^{\|S_n\|} p^i(1-p)(D_n^{(PCP)} - D_{n,i}^{(RCP)}) \end{aligned} \qquad (2)$$

Onde $E[\Delta D_n^i]$ é a redução de distorção esperada trazida por incluir a 1ª fatia redundante a partir de Sn. Além disso, $D_n^{(PCP)}$ é a distorção incorrida quando a fatia primária é perdida e Sn é um conjunto vazio.

Similarmente, $D_{n,i}^{(RCP)}$ é a distorção incorrida quando a 1ª fatia redundante codificada em S_n é corretamente decodificada, porém a fatia primária bem como I fatias redundantes

incluídas (i-1)ª no conjunto são perdidas.

A resolução diretamente do problema de otimização apresentado pela eq. (1) e (2) pode ser difícil. Em vez disso, um algoritmo de busca ávida é desenvolvido com baixa complexidade. Similar a outros algoritmos baseados em avidez, em cada etapa o algoritmo seleciona a melhor fatia redundante em termos da razão entre a redução de distorção e custo de taxa, até que a taxa de bits dada seja utilizada. Após uma fatia redundante ser selecionada, o algoritmo adiciona o mesmo ao conjunto como um elemento novo, ou substitui uma fatia redundante existente no conjunto com o novo se produzir redução de distorção esperada maior.

Considere P como uma fatia redundante candidata para a posição i de S_n para a imagem n. Observe que sua taxa de bits como $R_{n,P}^{(RCP}$ e seu correspondente $E[\Delta D_n^i]$ pode ser calculado como um termo na Eq. (2). Para cada conjunto S_n os requerentes atribuem um contador c para registrar o número de fatias redundantes que foram incluídas. Finalmente, indique $R_T^{(RCP)}$ como a taxa total de bits alocada a todas as fatias redundantes. As etapas detalhadas do algoritmo são listadas abaixo.

1. Inicialização: $\forall n \in [1, N]$, conjunto S_n para vazio e seu c em 0. Defina $R_T^{(RCP)}$ em $R_T - \sum_{n=1}^{N} R_n^{(PCP)}$.

2. Para todos os conjuntos S_n $\forall n \in [1, N]$, nas posições i ($i \in \{c, c+1\}$ e $i > 0$), selecione a fatia redundante P que tem a razão maior entre $E[\Delta D_n^i]$ e $R_{n,P}^{(RCP}$, entre todas as fatias candidatas nas posições.

3. Se $R_{n,P}^{(RCP} > R_T^{(RCP)}$, exclua a fatia redundante P como candidato para a posição i de S_n. Vá para a etapa 6.

4a. Se i == c + 1, inclua P na posição i de S_n e exclua o mesmo como candidato para a posição. Defina c em i e atualize $R_T^{(RCP)}$ para $R_T^{(RCP)} - R_{n,P}^{(RCP}$.

4b. Ou se (i ==c), o que significa a posição i de S_n já está ocupada por outra fatia P',

1) Se $E[\Delta D_n^i]$P > $E[\Delta D_n^i]$P', então substitua P' com P na posição. Atualize $R_T^{(RCP)}$ para $R_T^{(RCP)} + R_{n,P}^{(RCP} - R_{n,P}^{(RCP}$.

2) Exclua P como candidato para a posição.

5. Se houver outra fatia redundante candidata, vá para a etapa 2; de outro modo saia, e {S_n, $\forall n \in [1, N]$, contém o conjunto das fatias redundantes escolhidas para o fluxo de bits final.

Para ajudar a esclarecer a operação do algoritmo acima, o seguinte exemplo é fornecido no qual somente um conjunto único necessita ser cheio. No primeiro round, o algo-

ritmo avalia candidatos para a posição 1 do conjunto. Observe que os candidatos para todas as posições são iguais.

Durante o primeiro round, os requerentes assumirão que um candidato é selecionado na etapa 2 que também cumpre a etapa 3. A posição 1 é então como tentativa com esse candidato.

O algoritmo prossegue então para um segundo round no qual posições 1 e 2 do conjunto são avaliadas simultaneamente. Ao contrário do primeiro round, o segundo round pode envolver múltiplas passagens através do algoritmo.

No segundo round, o algoritmo determina, na etapa 2, o candidato com a melhor razão. Na determinação da melhor razão, o algoritmo avalia (1) todos os candidatos (exceto o candidato selecionado como tentativa para a posição 1) com base no valor esperado de redução de distorção para aposição 1, (2) todos os candidatos baseados no valor esperado de redução de distorção para a posição 2. O melhor desses "dois" conjuntos de candidatos é selecionado na etapa 2. O candidato selecionado pode ser para a posição 1 ou posição 2. Isso conclui a primeira passagem do segundo round.

Se o candidato recentemente selecionado for novamente para a posição 1, então os valores esperados de redução de distorção são comparados na etapa 4b para os candidatos que foram selecionados no primeiro round e o segundo round (primeira passagem). O candidato com o valor mais elevado (melhor) é selecionado como tentativa para a posição 1, desse modo possivelmente substituindo o candidato selecionado como tentativa no primeiro round. Além disso, o segundo round continua por executar uma segunda passagem através do algoritmo. Na segunda passagem, o algoritmo (na etapa 2) avalia as razões de (1) todos os candidatos (exceto os dois candidatos anteriormente selecionados para a posição 1) com base no valor esperado de redução de distorção para a posição 1, e (2) todos os candidatos com base no valor esperado de redução de distorção para a posição 2. Deve ser evidente que o segundo round pode exigir muitas passagens através do algoritmo. Em cada passagem através do algoritmo, o candidato selecionado mais recentemente (para a posição 1), juntamente com todos os outros candidatos anteriormente selecionados (para a posição 1) é eliminado de consideração adicional na avaliação de razões para a posição 1.

Entretanto, sempre que o candidato recentemente selecionado de qualquer passagem do segundo round for para a posição 2, então a posição 2 é cheia como tentativa com o candidato recentemente selecionado. Também a posição 1 é considerada como cheia porque o candidato (se selecionado durante o primeiro round ou o segundo round) não estará sujeito a substituição adicional. O algoritmo então prossegue para um terceiro round no qual as posições 2 e 3 são avaliadas simultaneamente.

Genericamente cada imagem pode ter um impacto diferente sobre a distorção de canal da sequência decodificada quando a imagem é perdida. Com o algoritmo proposto,

essas imagens com razões maiores entre $E\left[\Delta D_n^i\right]$ e $R_{n,P}^{(RCP}$ ou valores de $E\left[\Delta D_n^i\right]$ maiores ocupam mais posições e, portanto, recebem mais taxa de bits para suas fatias redundantes, consequentemente recebem proteção de erro mais forte. Isso forma proteção de erro desigual (UEP) através da sequência e é uma fonte do ganho de desempenho fornecido pelo algoritmo.

Como a importância de cada fatia redundante pode ser diferente, as fatias redundantes incluídas podem ser separadas de acordo com sua importância relativa. Portanto, é possível agrupar todas as fatias primárias para formar uma camada base, e dispor todas as fatias redundantes juntamente com importância decrescente em camadas de intensificação. Isso forma um fluxo de bits redimensionável em termos de resiliência de erro, isto é, melhor capacidade de resiliência de erro pode ser obtida incluindo mais camadas de intensificação do fluxo de bits. Por formar o fluxo de bits pré-codificado com redimensionamento de resiliência de erro, o fluxo de bits final pode ser obtido por simplesmente truncar os fluxos de bit pré-codificados de acordo com uma restrição de taxa. Simplifica o processo de montagem.

REIVINDICAÇÕES

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

receber uma solicitação para enviar através de um canal uma ou mais codificações de pelo menos uma porção de um objeto de dados (305);

acessar informações para determinar qual de múltiplas codificações pelo menos da porção do objeto de dados enviar através do canal (310); e

determinar, após receber a solicitação, um conjunto de codificações a enviar através do canal, o conjunto sendo determinado a partir das múltiplas codificações e incluindo pelo menos uma das múltiplas codificações, e o número de codificações no conjunto determinado sendo baseado nas informações acessadas (320).

2. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o conjunto determinado de codificações inclui uma ou mais codificações de fonte, e o número de codificações de fonte no conjunto determinado indica um nível específico de redundância de codificação de fonte.

3. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que pelo menos uma codificação no conjunto determinado codifica uma imagem em uma sequência de vídeo.

4. Método, de acordo com a reivindicação 3, **CARACTERIZADO** pelo fato de que pelo menos uma codificação no conjunto determinado é uma codificação de perda pelo menos da porção do objeto de dados.

5. Método, de acordo com a reivindicação 3, **CARACTERIZADO** pelo fato de que o conjunto determinado inclui uma codificação primária da imagem e uma codificação redundante da imagem.

6. Método, de acordo com a reivindicação 5, **CARACTERIZADO** pelo fato de que a codificação primária é uma imagem codificada primária e a codificação redundante é uma imagem codificada redundante.

7. Método, de acordo com a reivindicação 6, **CARACTERIZADO** pelo fato de que a imagem codificada primária e a imagem codificada redundante são compatíveis com o padrão H.264/AVC.

8. Método, de acordo com a reivindicação 3, **CARACTERIZADO** pelo fato de que compreende ainda:

determinar um segundo conjunto de codificações a enviar através do canal, o segundo conjunto sendo determinado a partir de múltiplas codificações de uma segunda imagem na sequência de vídeo, e o número de codificações no segundo conjunto determinado sendo baseado nas informações acessadas e possivelmente diferindo do número de codificações no conjunto determinado.

9. Método, de acordo com a reivindicação 3, **CARACTERIZADO** pelo fato de que

compreende ainda:

acessar segundas informações para determinar qual de múltiplas codificações de uma segunda imagem na sequência de vídeos enviar através do canal; e

determinar um segundo conjunto de codificações a enviar através do canal, o segundo conjunto sendo determinado a partir de múltiplas codificações da segunda imagem na sequência de vídeo, e o número de codificações no segundo conjunto determinado sendo baseado nas segundas informações acessadas e possivelmente diferindo do número de codificações no conjunto determinado.

10. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que compreende ainda armazenar as múltiplas codificações antes de receber a solicitação, e em que a determinação do conjunto compreende determinar o conjunto a partir das codificações armazenadas.

11. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o acesso às informações compreende acessar informações que descrevem uma condição de canal para o canal.

12. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que:

as informações acessadas compreendem capacidade disponível para o canal, e

a determinação do conjunto compreende determinar um conjunto de codificações que pode ser enviado através do canal na capacidade disponível.

13. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que:

as informações acessadas compreende taxa de erro para o canal; e

a determinação do conjunto compreende incluir um número relativamente menor de codificações no conjunto se a taxa de erro for mais baixa e incluir um número relativamente maior de codificações no conjunto se a taxa de erro for mais elevada.

14. Método, de acordo com a reivindicação 13, **CARACTERIZADO** pelo fato de que:

as múltiplas codificações incluem múltiplas fatias redundantes para uma imagem dada em uma sequência de vídeo, e

a determinação do conjunto compreende ainda incluir um número relativamente menor de fatias redundantes múltiplas no conjunto se a taxa de erro for mais baixa e incluir um número relativamente maior das múltiplas fatias redundantes no conjunto se a taxa de erro for mais elevada.

15. Método, de acordo com a reivindicação 14, **CARACTERIZADO** pelo fato de que:

as múltiplas fatias redundantes são compatíveis com o padrão H.264/AVC,

o conjunto determinado inclui pelo menos uma das fatias redundantes, e

o método compreende ainda enviar o conjunto determinado de codificações, inclu-

indo pelo menos uma fatia redundante, a um receptor em uma forma compatível com o padrão H.264/AVC.

16. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que as informações acessadas compreendem uma ou mais entre (1) capacidade disponível para o canal, (2) taxa de erro para o canal, e (3) custo para transmitir através do canal.

17. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que:

as múltiplas codificações estão em um conjunto ordenado de codificações;

cada codificação no conjunto ordenado codifica pelo menos a porção do objeto de dados,

as codificações no conjunto ordenado são ordenadas de acordo com uma métrica relacionada à qualidade da codificação comparada com dados originais codificados pela codificação, e

a determinação do conjunto de codificações compreende determinar uma codificação alvo no conjunto ordenado e incluir no conjunto todas as codificações a partir de um ponto final do conjunto ordenado até e incluindo a codificação alvo.

18. Método, de acordo com a reivindicação 17, **CARACTERIZADO** pelo fato de que:

o canal é um canal de perda, e

as codificações no conjunto ordenado são ordenadas de tal modo que após uma codificação específica no conjunto ordenado ter sido enviada para um dispositivo através do canal de perda, se a codificação seguinte que ocorre após a codificação específica no conjunto ordenado for também enviada ao dispositivo através do canal de perda, uma qualidade esperada de uma decodificação pelo dispositivo pelo menos da porção do objeto de dados aumenta.

19. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que compreende ainda duplicar pelo menos uma das múltiplas codificações, e em que a determinação do conjunto compreende incluir a codificação duplicada no conjunto.

20. Meio legível por computador **CARACTERIZADO** por compreender instruções para fazer com que um ou mais dispositivo execute o que se segue:

receber (305) uma solicitação para enviar através de um canal uma ou mais codificações pelo menos de uma porção de um objeto de dados;

acessar (310) informações para selecionar qual das múltiplas codificações pelo menos da porção do objeto de dados a enviar através do canal; e

selecionar, após receber a solicitação, um conjunto de codificações a enviar através do canal, o conjunto sendo determinado a partir das múltiplas codificações e incluindo pelo menos uma das múltiplas codificações, e o número de codificações no conjunto determinado sendo baseado nas informações acessadas (230).

21. Aparelho, **CARACTERIZADO** por compreender:

meio (130) para acessar informações para selecionar qual das múltiplas codificações pelo menos de uma porção de um objeto de dados a enviar através de um canal; e

meio (130) para selecionar um conjunto de codificações a enviar através do canal, o conjunto sendo determinado a partir das múltiplas codificações e incluindo pelo menos uma das múltiplas codificações, e o número de codificações no conjunto determinado sendo baseado nas informações acessadas.

22. Unidade de seleção (130), **CARACTERIZADA** por ser configurada para acessar informações para determinar qual das múltiplas codificações de pelo menos uma porção de um objeto de dados enviar através de um canal, e determinar um conjunto de codificações para enviar através do canal, o conjunto sendo determinado a partir das múltiplas codificações e incluindo pelo menos uma das múltiplas codificações, e o número de codificações no conjunto determinado sendo baseado nas informações acessadas.

23. Método, **CARACTERIZADO** por compreender:

fornecer (410) informações para determinar qual das múltiplas codificações pelo menos de uma porção de um objeto de dados enviar através de um canal; e

receber (420) um conjunto de codificações através do canal, o conjunto tendo sido determinado a partir das múltiplas codificações e incluindo pelo menos uma das codificações múltiplas, e o número de codificações no conjunto tendo sido baseado nas informações fornecidas.

24. Método, de acordo com a reivindicação 23, **CARACTERIZADO** pelo fato de que:

as informações fornecidas compreendem informações que descrevem uma condição de canal do canal, e

o método compreende ainda determinar as informações que descrevem a condição de canal com base nos dados recebidos através do canal.

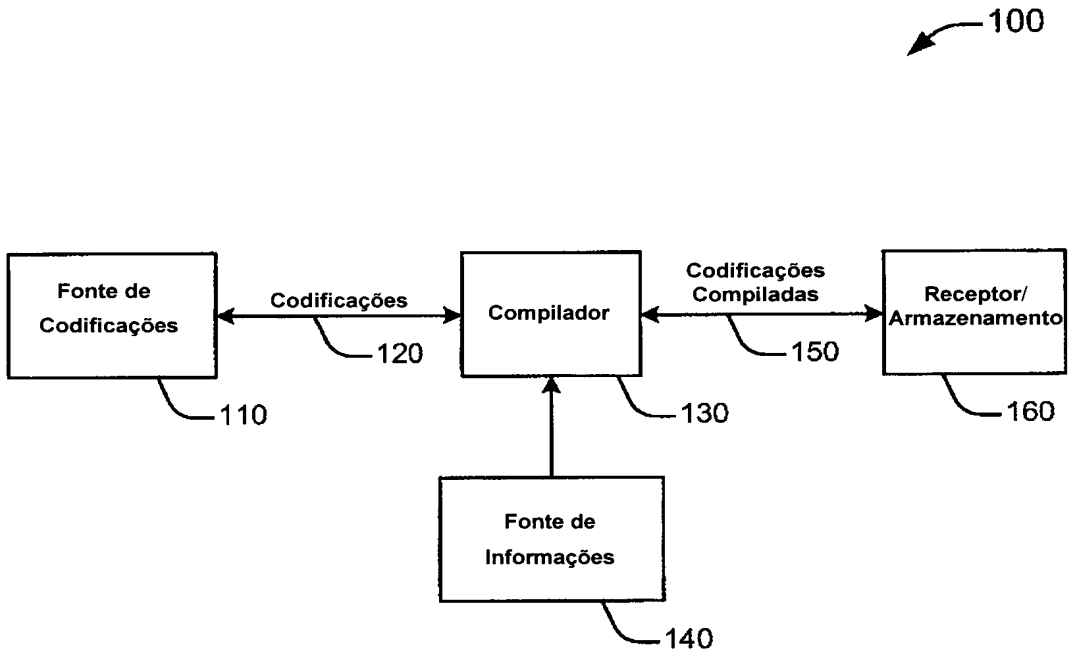

100
Fonte de Codificações
Codificações
120
Compilador
Codificações Compiladas
150
Receptor/ Armazenamento
110
130
160
Fonte de Informações
140

FIG. 1

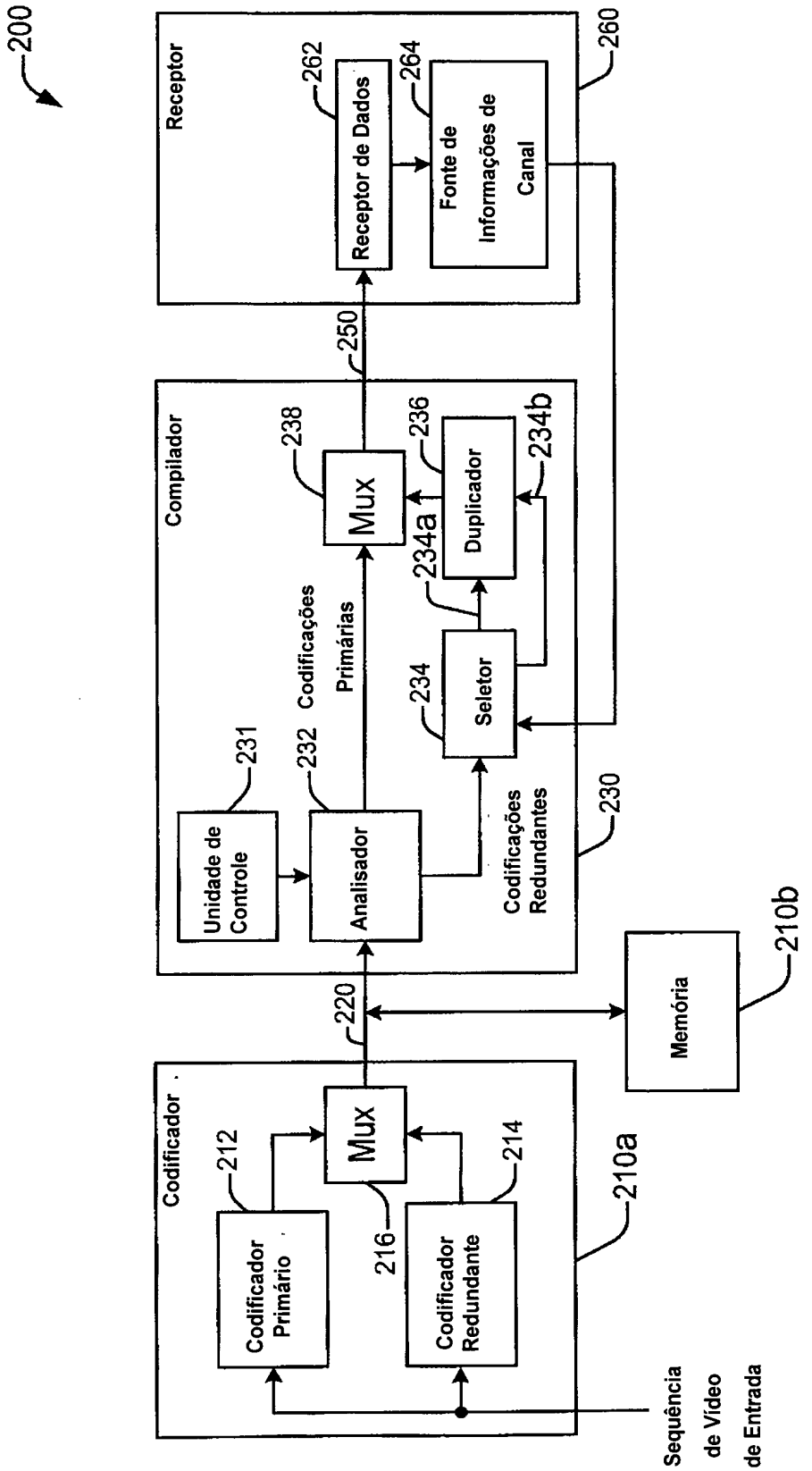

200
Receptor
262
Receptor de Dados
264
Fonte de Informações de Canal
260
250
Compilador
238
Mux
236
Duplicador
234b
234a
Codificações Primárias
234
Seletor
231
Unidade de Controle
232
Analisador
Codificações Redundantes
230
220
Memória
210b
Codificador
212
Codificador Primário
Mux
214
216
Codificador Redundante
210a
Sequência de Vídeo de Entrada

FIG.2

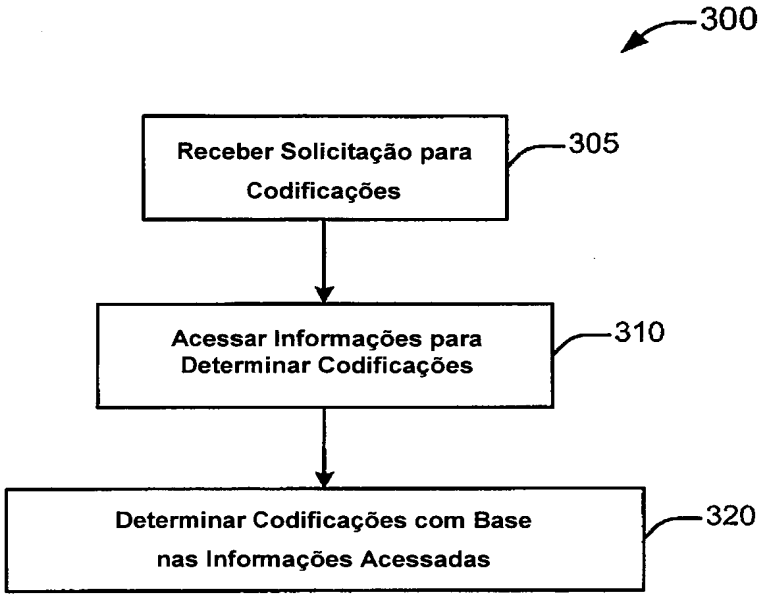

300
Receber Solicitação para Codificações
305
Acessar Informações para Determinar Codificações
310
Determinar Codificações com Base nas Informações Acessadas
320

FIG.3

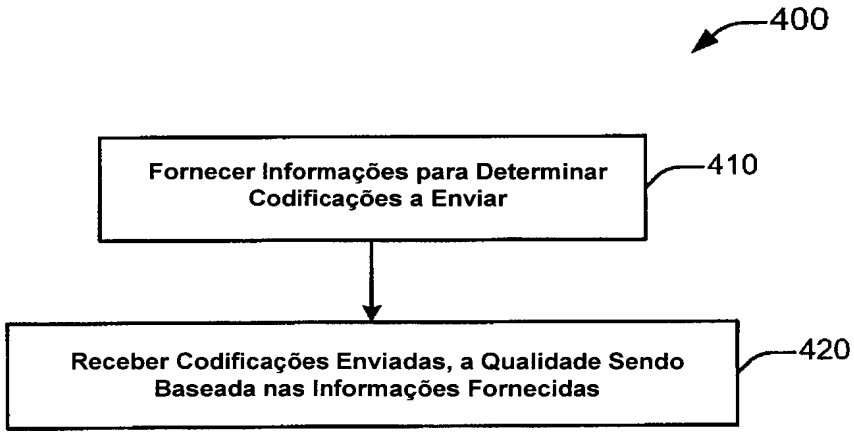

400
Fornecer Informações para Determinar Codificações a Enviar
410
Receber Codificações Enviadas, a Qualidade Sendo Baseada nas Informações Fornecidas
420

FIG.4

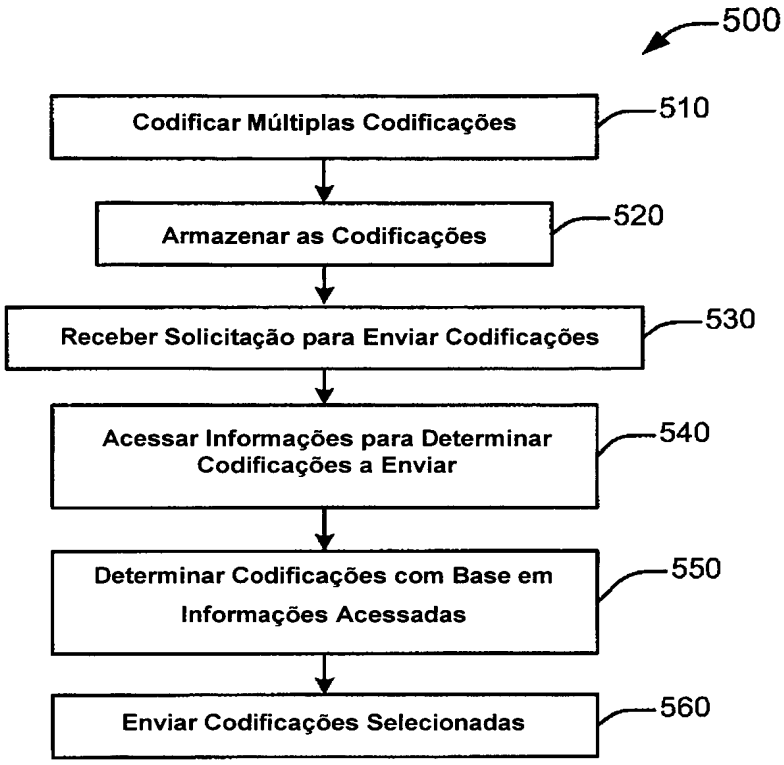

500
Codificar Múltiplas Codificações
510
Armazenar as Codificações
520
Receber Solicitação para Enviar Codificações
530
Acessar Informações para Determinar Codificações a Enviar
540
Determinar Codificações com Base em Informações Acessadas
550
Enviar Codificações Selecionadas
560

FIG.5

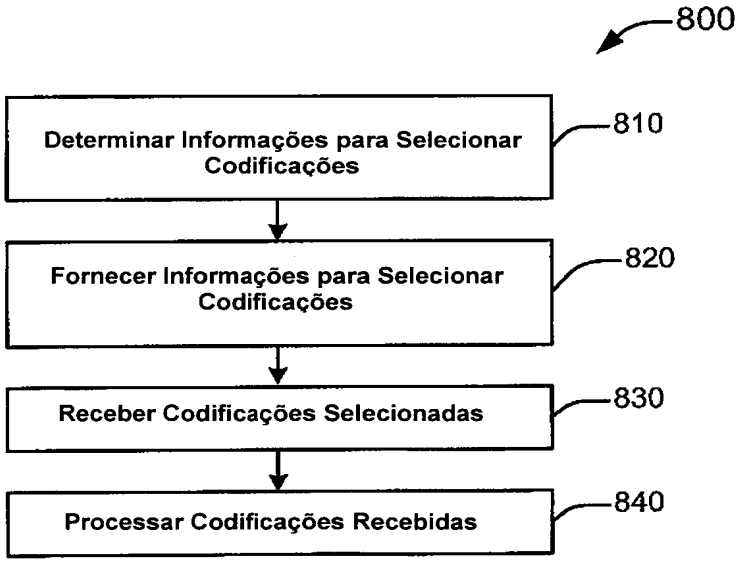

800
Determinar Informações para Selecionar Codificações
810
Fornecer Informações para Selecionar Codificações
820
Receber Codificações Selecionadas
830
Processar Codificações Recebidas
840

FIG.8

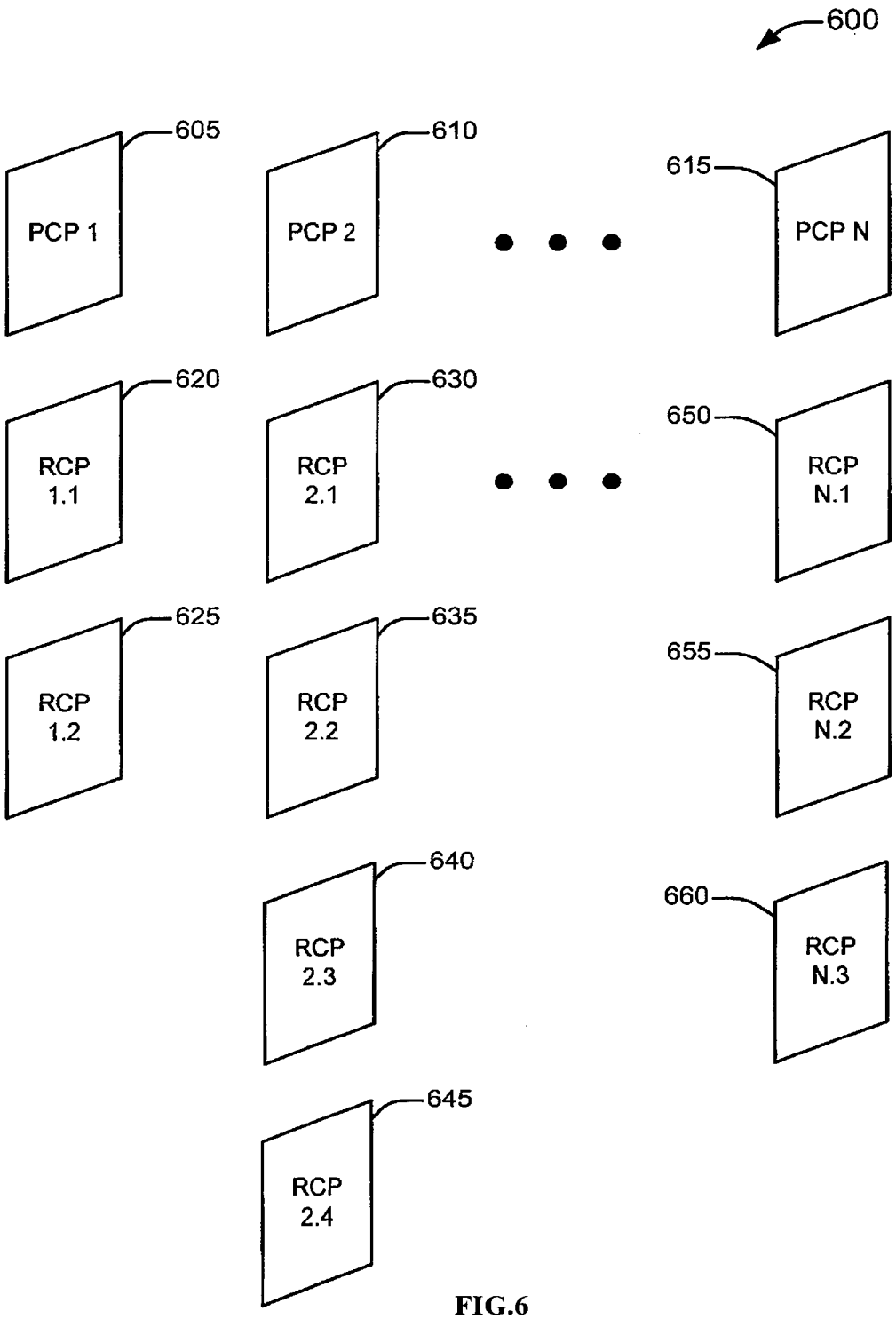

600
605
PCP 1
610
PCP 2
615
PCP N
620
RCP 1.1
630
RCP 2.1
650
RCP N.1
625
RCP 1.2
635
RCP 2.2
655
RCP N.2
640
RCP 2.3
660
RCP N.3
645
RCP 2.4

FIG.6

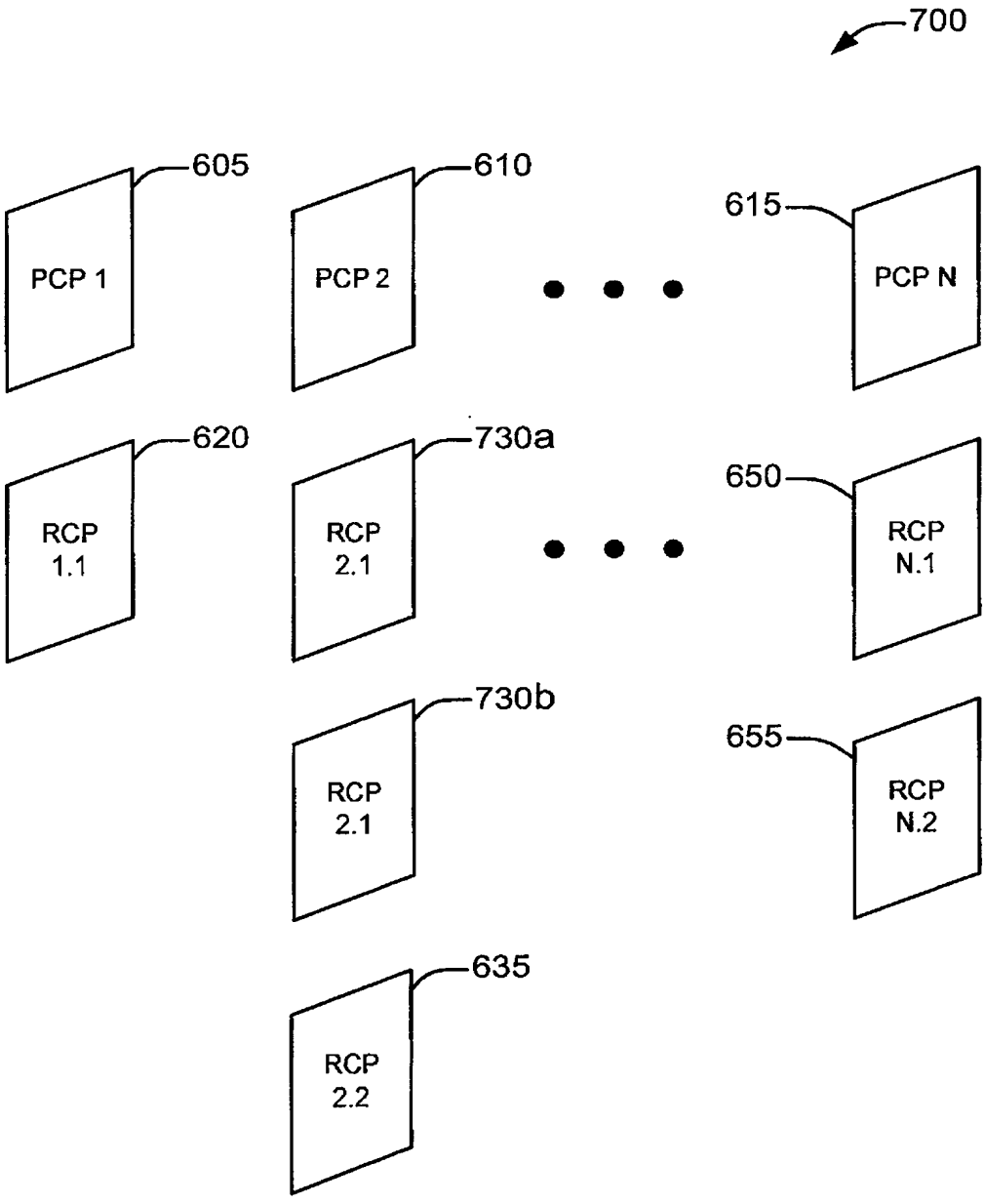

700
605
PCP 1
610
PCP 2
615
PCP N
620
RCP 1.1
730a
RCP 2.1
650
RCP N.1
730b
RCP 2.1
655
RCP N.2
635
RCP 2.2

FIG.7

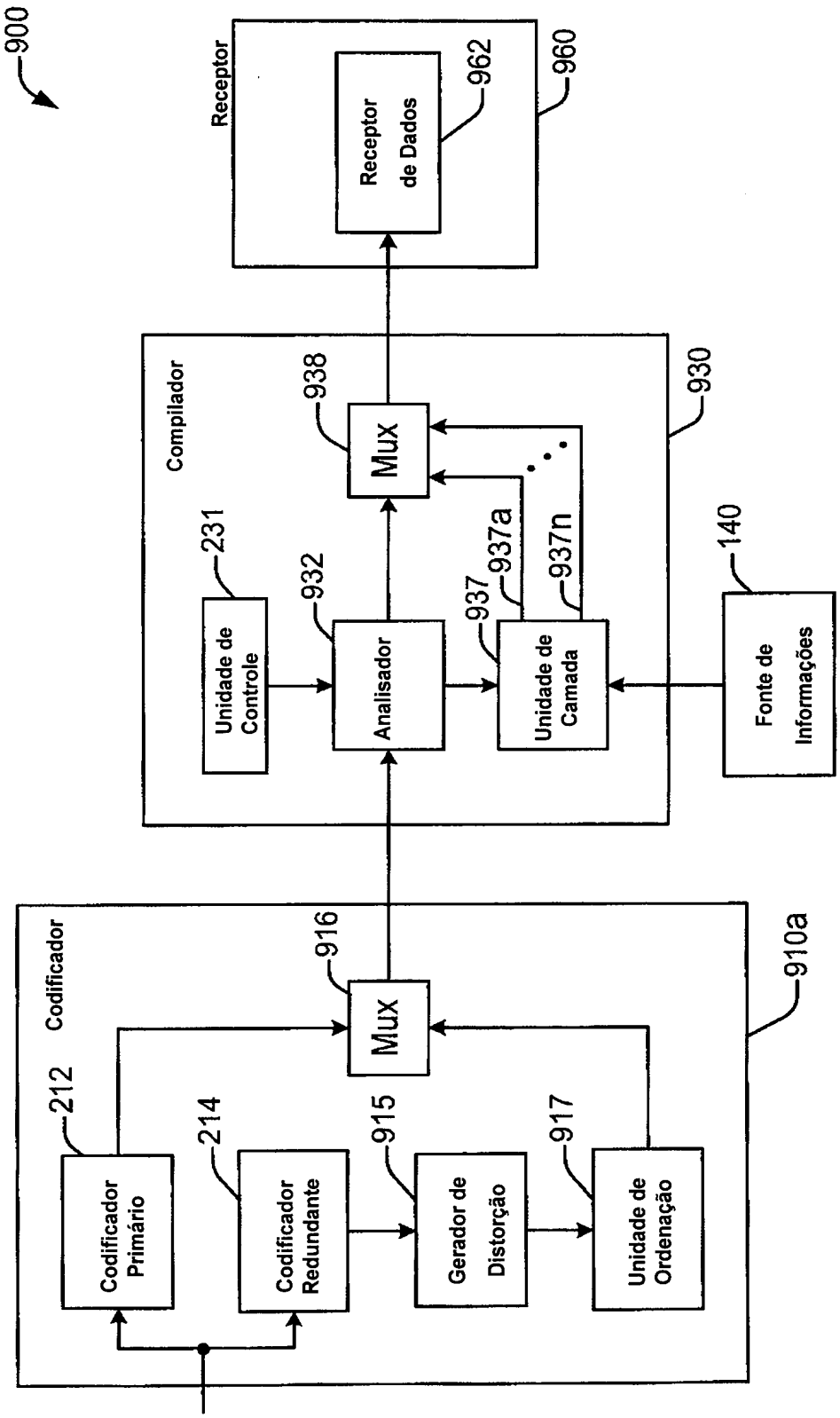

900
Receptor
Receptor de Dados
962
960
Compilador
938
Mux
930
231
Unidade de Controle
932
Analisador
937
937a
937n
Unidade de Camada
140
Fonte de Informações
Codificador
916
Mux
910a
212
Codificador Primário
214
Codificador Redundante
915
Gerador de Distorção
917
Unidade de Ordenação

FIG. 9

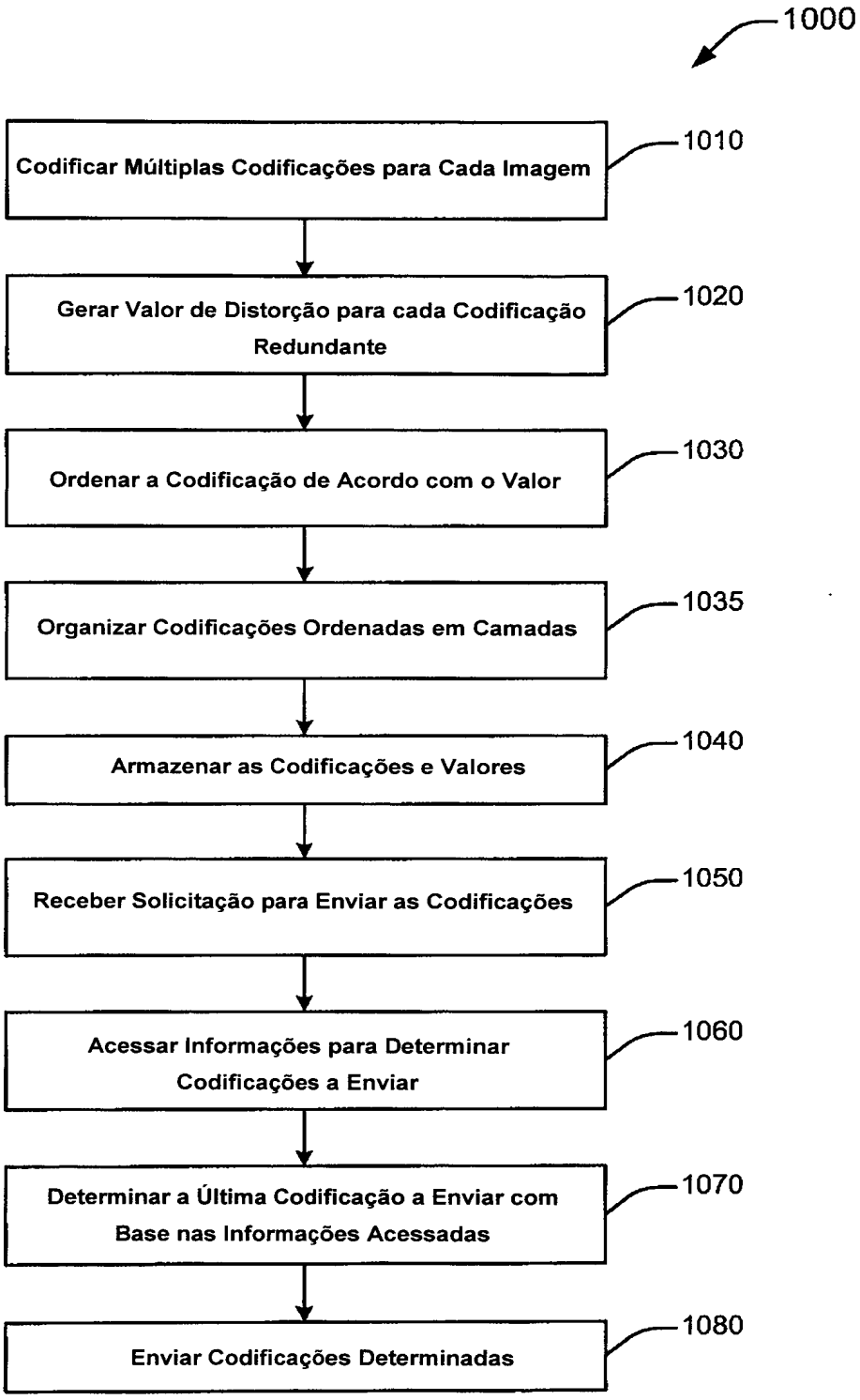

1000
Codificar Múltiplas Codificações para Cada Imagem
1010
Gerar Valor de Distorção para cada Codificação Redundante
1020
Ordenar a Codificação de Acordo com o Valor
1030
Organizar Codificações Ordenadas em Camadas
1035
Armazenar as Codificações e Valores
1040
Receber Solicitação para Enviar as Codificações
1050
Acessar Informações para Determinar Codificações a Enviar
1060
Determinar a Última Codificação a Enviar com Base nas Informações Acessadas
1070
Enviar Codificações Determinadas
1080

FIG. 10

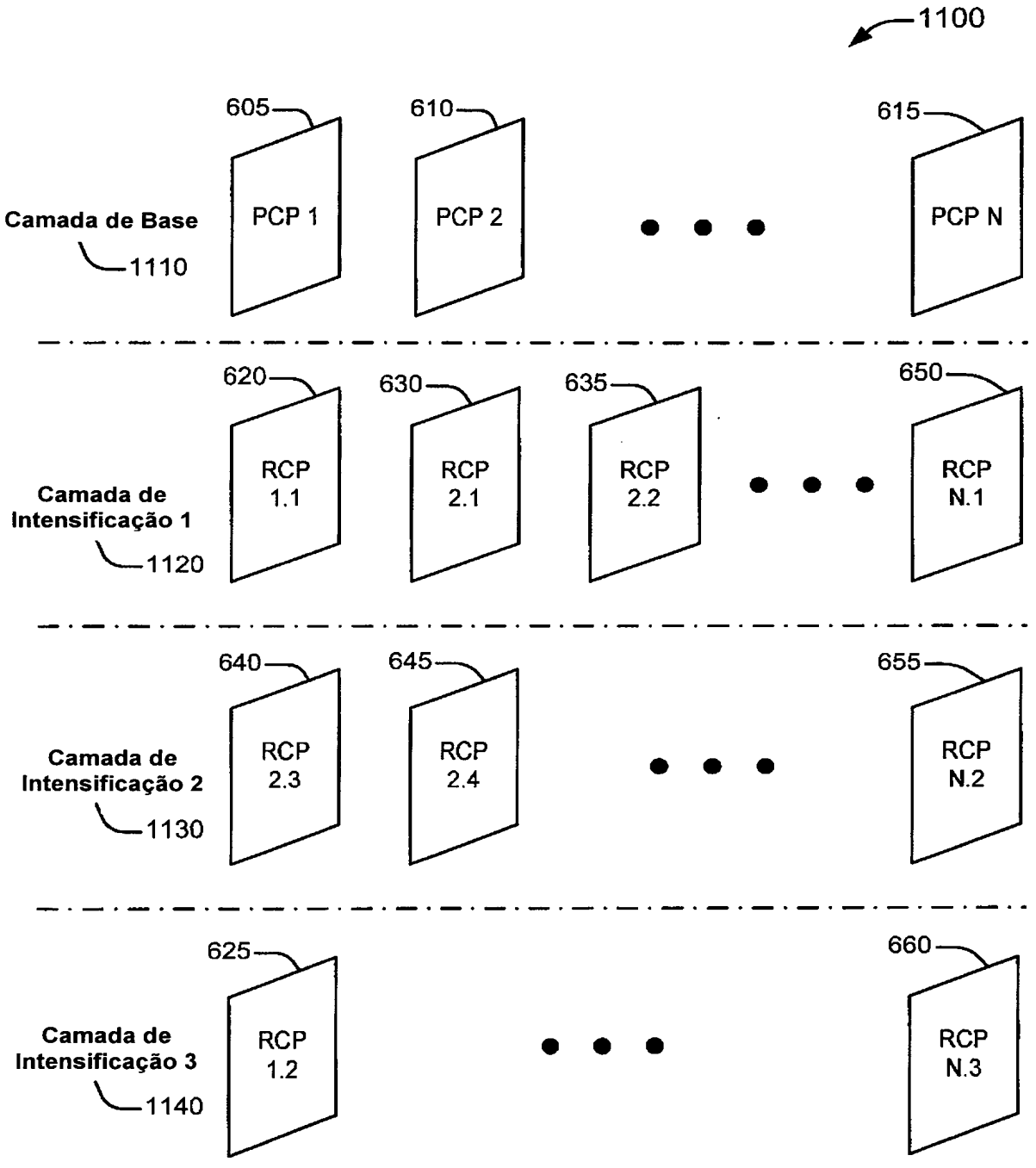

1100
605
610
615
Camada de Base
1110
PCP 1
PCP 2
PCP N
620
630
635
650
Camada de Intensificação 1
1120
RCP 1.1
RCP 2.1
RCP 2.2
RCP N.1
640
645
655
Camada de Intensificação 2
1130
RCP 2.3
RCP 2.4
RCP N.2
625
660
Camada de Intensificação 3
1140
RCP 1.2
RCP N.3

FIG. 11

RESUMO

"CODIFICAÇÃO DE REDUNDÂNCIA FLEXÍVEL"

Várias implementações reveladas permitem que uma quantidade flexível de redundância seja utilizada em codificação. Em uma implementação geral, informações são acessadas para determinar qual das múltiplas codificações pelo menos de uma porção de um objeto de dados deve enviar através de um canal (310). Um conjunto de múltiplas codificações é determinado para enviar através do canal, com o conjunto incluindo pelo menos um e possivelmente mais das múltiplas codificações, e o número de codificações no conjunto sendo baseado nas informações acessadas (320). Em uma implementação mais específica, a característica de fatia redundante do padrão de codificação H.264/AVC é utilizada, e um número variável de fatias redundantes é transmitido para qualquer imagem dada com base em condições atuais de canal.