



República Federativa do Brasil
Ministério da Economia
Instituto Nacional da Propriedade Industrial

(11) BR 122021000421-8 B1



(22) Data do Depósito: 03/12/2008

(45) Data de Concessão: 18/01/2022

(54) Título: REFERENCIAÇÃO DE SUB-FLUXO FLEXÍVEL DENTRO DE UM FLUXO DE DADOS DE TRANSPORTE

(51) Int.Cl.: H04N 21/2343; H04N 21/2662; H04N 21/43; H04N 21/845.

(52) CPC: H04N 21/234327; H04N 21/2662; H04N 21/4305; H04N 21/8455.

(30) Prioridade Unionista: 25/04/2008 EP PCT/EP2008/003384.

(73) Titular(es): FRAUNHOFER-GESELLSCHAFT ZUR FÖRDERUNG DER ANGEWANDTEN FORSCHUNG E.V..

(72) Inventor(es): THOMAS SCHIERL; CORNELIUS HELLGE; KARSTEN GRÜNEBERG.

(86) Pedido PCT: PCT EP2008010258 de 03/12/2008

(87) Publicação PCT: WO 2009/129838 de 29/10/2009

(85) Data do Início da Fase Nacional: 11/01/2021

(62) Pedido Original do Dividido: PI0822167-7 - 03/12/2008

(57) Resumo: A presente invenção refere-se a uma representação de uma sequência de vídeo que apresenta um primeiro fluxo de dados compreendendo primeira porções de dados, sendo que as primeiras porções de dados compreendem primeira informação de sincronização e um segundo fluxo de dados, sendo que o segundo fluxo de dados compreende uma segunda porção de dados que apresenta segunda informação de sincronização, que pode ser derivada. Informação de associação é associada a uma segunda porção de dados do segundo fluxo de dados, sendo que a informação de associação indica uma primeira porção de dados pré-determinada do primeiro fluxo de dados. É gerado um fluxo de transporte que compreende o primeiro e o segundo fluxo de dados como a representação da sequência de vídeo.

"REFERENCIAÇÃO DE SUB-FLUXO FLEXÍVEL DENTRO DE UM FLUXO DE DADOS DE TRANSPORTE"

PEDIDO DIVIDIDO DO PI 0822167-7 depositado em 03/12/2008.

DESCRIÇÃO

[0001] Concretização da presente invenção diz respeito a esquemas para referenciar flexivelmente porções de dados individuais de diferentes sub-fluxos de um fluxo de dados de transporte, contendo dois ou mais sub-fluxos. Particularmente, diversas concretizações referem-se a um método e a um equipamento para identificar porções de dados de referência contendo informação sobre imagens de referência necessárias para a decodificação de um fluxo de vídeo de uma camada superior de um fluxo de vídeo escalável quando fluxos de vídeo com diferentes propriedades de sincronização são combinados em um único fluxo de transporte.

[0002] São inúmeras as aplicações, nas quais fluxos de dados múltiplos são combinados dentro de um fluxo de transporte. Essa combinação ou multiplexação dos diferentes fluxos de dados é muitas vezes necessária para possibilitar a transmissão da informação completa utilizando apenas um único canal de transporte físico para transmitir o fluxo de transporte gerado.

[0003] Por exemplo, em um fluxo de transporte MPEG-2 usado para transmissão por satélite de múltiplos programas de vídeo, sendo que cada programa de vídeo fica contido dentro de um fluxo elementar. Isto é, frações de dados de um fluxo elementar específico (que são empacotadas nos assim chamados

pacotes PES) são intercaladas com frações de dados de outros fluxos elementares. Além disso, diferentes fluxos elementares ou sub-fluxos podem pertencer a um único programa como, por exemplo, o programa pode ser transmitido utilizando-se um fluxo elementar de áudio e um fluxo elementar de vídeo separado. Os fluxos elementares de áudio e de vídeo são, portanto, dependentes entre si. Ao usar códigos de vídeo escaláveis (SVC), as interdependências podem ser ainda mais complicadas, como um vídeo da camada base AVC (Advanced Video Codec) com compatibilidade regressiva (H.264/AVC) pode ser então melhorado pela adição de informação adicional, os assim chamados sub-fluxos de bits SVC, que melhoram a qualidade da camada base AVC em termos de fidelidade, resolução espacial e/ou resolução temporal. Isto é, nas camadas avançadas (os sub-fluxos de bits SVC), pode ser transmitida informação adicional para um quadro de vídeo para melhorar sua qualidade perspectiva.

[0004] Para a reconstrução, toda informação pertencente a um único quadro de vídeo é coletada dos diferentes fluxos antes de uma decodificação do respectivo quadro de vídeo. A informação contida dentro de diferentes fluxos que pertence a um único quadro é chamada de unidade NAL (Network Abstraction Layer Unit). A informação pertencente a uma única imagem pode ser ainda transmitida por diferentes canais de transmissão. Por exemplo, um canal físico separado pode ser usado para cada sub-fluxo de bits. Porém, os diferentes pacotes de dados dos sub-fluxos de bits individuais dependem um do outro. A dependência é muitas

vezes sinalizada por um elemento de sintaxe específico (dependência_ID: DID) da sintaxe de fluxo de bits. Isto é, os sub-fluxos de bits SVC (diferindo no elemento de sintaxe no cabeçalho de unidade H.264/SVC NAL: DID), que melhoram a camada base AVC ou um sub-fluxo de bits inferior em pelo menos uma das dimensões de escalabilidade possíveis fidelidade, resolução espacial ou temporal, são transportados para o fluxo de transporte com diferentes números PID (identificador de pacote). Eles são, por assim dizer, transportados do mesmo modo como diferentes tipos de mídia (p.ex. áudio ou vídeo) para o mesmo programa que deverá ser transportado. A presença desses sub-fluxos é definida em um cabeçalho de pacote de fluxo de transporte associado ao fluxo de transporte.

[0005] Porém, para reconstruir e decodificar as imagens e os dados de áudio associados, os diferentes tipos de mídia devem ser sincronizados antes ou após a decodificação. A sincronização após a decodificação é muitas vezes obtida pela transmissão das assim chamadas “estampas de tempo de apresentação” (PTS) que indicam o atual tempo de saída/apresentação tp de um quadro de vídeo ou um quadro de áudio, respectivamente. Se um buffer de imagem decodificada (DPB) for usado para armazenar temporariamente uma imagem decodificada (quadro) de um fluxo de vídeo transportado após a decodificação, a estampa de tempo de apresentação tp portanto indicará a retirada da imagem decodificada do buffer respectivo. Como diferentes tipos de quadro podem ser usados, por exemplo, quadros tipo p (predictivo) e tipo b (bi-

direcional), os quadros de vídeo não precisam ser necessariamente decodificados para que sejam apresentados. Portanto, as assim chamadas “estampas de tempo de decodificação” são normalmente transmitidas, as quais indicam o último tempo possível de decodificação de um quadro para garantir assim que toda a informação esteja presente para os quadros subsequentes.

[0006] Quando a informação recebida do fluxo de transporte é temporariamente armazenada dentro de um buffer de fluxo elementar (EB), a estampa de tempo de decodificação (DTS) indica o último tempo possível de retirada da informação em questão do buffer de fluxo elementar (EB). O processo de decodificação convencional pode, portanto, ser definido em termos de um modelo de armazenamento temporário hipotético (T-STD) para a camada de sistema e um modelo de armazenamento temporário (HRD) para a camada de vídeo. Entende-se por camada de sistema a camada de transporte, isto é, uma sincronização precisa da multiplexação e demultiplexação necessária para prover diferentes fluxos de programa ou fluxos elementares dentro de um único fluxo de transporte é vital. Entende-se por camada de vídeo a informação de divisão em pacotes e referência exigida pelo codec de vídeo empregado. As informações dos pacotes de dados da camada de vídeo são novamente empacotadas e combinadas pela camada de sistema para permitir assim uma transmissão em série do canal de transporte.

[0007] Um exemplo de um modelo de armazenamento temporário hipotético utilizado pela transmissão de vídeo

MPEG-2 com um único canal de transporte é apresentado na figura 1. As estampas de tempo da camada de vídeo e as estampas de tempo da camada de sistema (indicadas no cabeçalho PES) indicam o mesmo instante de tempo. Se, porém, a frequência de sincronização da camada de vídeo e da camada de sistema for diferente (como normalmente é o caso), os tempos serão iguais dentro da tolerância mínima indicada pelos diferentes relógios usados pelos dois diferentes modelos de buffer (STD e HRD).

[0008] No modelo descrito pela figura 1, um pacote 2 de dados de fluxo de transporte que chega em um receptor no instante de tempo $t(i)$ é de-multiplexado do fluxo de transporte para diferentes fluxos independentes 4a - 4d, sendo que os diferentes fluxos são diferenciados por diferentes números PID presentes dentro de cada cabeçalho de pacote de fluxo de transporte.

[0009] Os pacotes de dados de fluxo de transporte são armazenados em um buffer de transporte 6 (TB) e então transferidos para um buffer de multiplexação 8 (MB). A transferência do buffer de transporte TB para o buffer de multiplexação MB pode ser feita com uma taxa fixada.

[0010] Antes de transmitir os dados puros de vídeo para um decodificador de vídeo, a informação adicional adicionada pela camada de sistema (camada de transporte), o cabeçalho PES é removido. Isso pode ser feito antes da transferência dos dados para um buffer de fluxo elementar 10 (EB). Isto é, a informação de sincronização correspondente removida, como por exemplo, a estampa de tempo de

decodificação t_d e/ou a estampa de tempo de apresentação t_p , deverá ser armazenada como informação lateral para processamento adicional quando os dados são transferidos do MB para EB. Para levar em conta uma reconstrução aceitável, os dados de unidade de acesso $A(j)$ (os dados correspondem a um quadro específico) são removidos não depois de $t_d(j)$ do buffer 10 de fluxo elementar 10, conforme indicado pela estampa de tempo de decodificação inserida no cabeçalho PES. Novamente, enfatizamos o fato de que a estampa de tempo de decodificação da camada de sistema deve ser igual à estampa de tempo de decodificação na camada de vídeo, já que as estampas de tempo de decodificação da camada de vídeo (indicadas pelas assim chamadas mensagens SEI para cada unidade de acesso $A(j)$) não são enviadas em texto puro dentro da sequência de bits de vídeo. Portanto, ao utilizar as estampas de decodificação da camada de vídeo, seria necessária ainda a decodificação do fluxo de vídeo e portanto, isso tornaria uma implementação multiplexada simples e eficiente impraticável.

[0011] Um decodificador 12 decodifica o conteúdo de vídeo puro para prover uma imagem decodificada, que é armazenada em um buffer de imagem decodificada 14. Conforme acima indicado, a estampa de tempo de apresentação provida pelo codec de vídeo é utilizada para controlar a apresentação, isto é, a remoção do conteúdo armazenado no buffer de imagem decodificada 14 (DPB).

[0012] Conforme anteriormente ilustrado, o presente padrão para o transporte de códigos de vídeo escaláveis (SVC)

define o transporte dos sub-fluxos de bits como fluxos elementares que apresentam pacotes de fluxo de transporte com diferentes números PID. Isso exige uma reordenação adicional dos dados de fluxo elementar contidos nos pacotes de fluxo de transporte para derivar as unidades de acesso individuais que representam um quadro único.

[0013] O esquema de reordenação aparece ilustrado na figura 2. O de-multiplexador 4 de-multiplexa pacotes que apresentam diferentes números PID em cadeias de buffer separadas de 20a a 20c. Ou seja, quando um fluxo de vídeo SVC é transmitido, partes de uma unidade de acesso idênticas transportadas para diferentes sub-fluxos são providas para diferentes buffers de representação de dependência (DRBn) de diferentes cadeias de buffer de 20a a 20c. Finalmente, estes devem ser previstos para um buffer 10 (EB) de fluxo elementar comum 10 (EB), que armazena temporariamente os dados antes de serem providos para o decodificador 22. A imagem decodificada é então armazenada em um buffer 24 de imagem decodificada comum.

[0014] Em outras palavras, partes da mesma unidade de acesso nos diferentes sub-fluxos de bits (que também são chamados de representações de dependência DR) são previamente armazenadas em buffers de representação de dependência (DRB) até que elas possam ser transmitidas ao buffer 10 (EB) de fluxo elementar para remoção. Um sub-fluxo de bits com o elemento de sintaxe máximo "dependência_ID" (DID), que aparece indicado dentro do cabeçalho da unidade NAL, compreende todas as unidades de acesso ou partes das unidades

de acesso (ou seja das representações de dependência DR) com a taxa de quadro máxima. Por exemplo, um sub-fluxo que é identificado pela dependência ID = 2, pode conter informação de imagem codificada com uma taxa de quadro de 50Hz, enquanto o sub-fluxo com dependência _ID = 1 pode conter informação para uma taxa de quadro de 25Hz.

[0015] De acordo com as presentes implementações, todas as representações de dependência dos sub-fluxos de bits com tempos iguais de decodificação são transmitidas para o decodificador na qualidade uma unidade de acesso particular da representação de dependência com o valor disponível máximo de DID. Ou seja, quando a representação de dependência com DID = 2 é decodificada, são consideradas informações de representações de dependência com DID = 1 e DID = 0. A unidade acesso é formada utilizando-se todos os pacotes de dados das três camadas que apresentam uma estampa de tempo de decodificação td idêntica. A ordem na qual as diferentes representações de dependência são providas para o decodificador, é definida pelo DID dos sub-fluxos considerados. A de-multiplexação e reordenação é realizada conforme indicado na figura 2. Uma unidade de acesso é abreviada com a letra A. DBP indica um buffer de imagem decodificada e DR indica uma representação de dependência. As representações de dependências são temporariamente armazenadas em buffer de representação de dependência DRB e o fluxo re-multiplexado é armazenado em um buffer de fluxo elementar EB antes da transmissão para o decodificador 22. MB significa buffers de multiplexação e PID significa o programa

ID de cada sub-fluxo individual. TB indica os buffers de transporte e td indica a estampa de tempo de codificação.

[0016] Porém, a abordagem anteriormente descrita sempre pressupõe o fato de a mesma informação de sincronização estar presente dentro de todas as representações de dependência dos sub-fluxos de bits associados à mesma unidade de acesso (quadro). Isso, porém, pode não ser verdade ou viável com conteúdo SVC, nem para as estampas de tempo de decodificação nem para as estampas de tempo de apresentação suportadas pelos sincronismos SVC.

[0017] Este problema pode surgir, já que anexo A do padrão H.264/AVC define diversos diferentes perfis e níveis. Geralmente, um perfil define as características que um decodificador em conformidade com aquele perfil particular, deve suportar. Os níveis definem o tamanho dos diferentes buffers dentro do decodificador. Além disso, os assim chamados “decodificadores hipotéticos de referência” (HRD) são definidos como um modelo que simula o comportamento desejado do decodificador, especialmente dos buffers associados no nível selecionado. O modelo HRD também é usado no codificador para assegurar que a informação de sincronização introduzida no fluxo de vídeo codificado pelo codificador não interrompe as restrições do modelo HRD e, conseqüentemente, o tamanho do buffer no decodificador. Isso, portanto tornaria impossível a decodificação com um decodificador compatível padrão. Um fluxo SVC pode suportar diferentes níveis dentro de diferentes sub-fluxos. Ou seja, a extensão SVC para codificação de vídeo apresenta a

possibilidade de criar diferentes sub-fluxos com diferente informação de sincronização. Por exemplo, diferentes taxas de quadro podem ser codificadas dentro dos sub-fluxos individuais de um fluxo de vídeo SVC.

[0018] A extensão escalável de H.264/AVC (SVC) possibilita a codificação de fluxos escaláveis com diferentes taxas de quadro em cada sub-fluxo. As taxas de quadro podem ser um múltiplo de cada uma, por exemplo camada de base 15Hz e camada de otimização temporal 30Hz. Além disso, SVC também permite ter uma razão quadro-taxa alterada entre os sub-fluxos, por exemplo, a camada de base provê 25 Hz e a camada de otimização 30 Hz. Note que o padrão SVC estendido ITU-T H.222.0 (camada de sistema) asystem-layer) é capaz de suportar tais estruturas de codificação.

[0019] A Fig. 3 mostra um exemplo de diferentes taxas de quadro dentro de dois sub-fluxos de um fluxo video de transporte. A camada de base (o primeiro fluxo de dados) 40 pode apresentar uma taxa de quadro de 30Hz e a camada de otimização temporal 42 do canal 2 (o segundo fluxo de dados) podem apresentar uma taxa de quadro de 50Hz. Para a camada de base, a informação de sincronização (DTS e PTS) no cabeçalho PES do fluxo de transporte ou o sincronismo nas SEIs do fluxo de vídeo são suficientes para decodificar a taxa de quadro mais baixa da camada de base.

[0020] Se a informação completa de um quadro de vídeo tiver sido incluída nos pacotes de dados da camada de otimização, a informação de sincronização nos cabeçalhos PES ou nos SEIs em-fluxo na camada de otimização, então serão

suficientes para a decodificação das taxas máxima de quadro. Como, porém, MPEG prove mecanismos complexos de referenciação mediante introdução de quadros-p ou quadros-i, pacotes de dados da camada de otimização podem utilizar da camada base como quadros de referência. Isto é, um quadro decodificado da camada de otimização utiliza informação em quadros previstos pela camada base. Essa situação aparece ilustrada na figura 3, onde as duas porções de dados ilustradas 40a e 40b da camada base 40 apresentam estampas de tempo de decodificação correspondentes ao tempo de apresentação para atender as exigências do modelo HRD para decodificadores de camada base bastante lentos. A informação necessária para um decodificador de camada de otimização para a completa decodificação de um quadro completo é fornecida pelos blocos de dados 44a a 44d.

[0021] O primeiro quadro 44a a ser reconstruído com uma taxa de quadro superior exige a informação completa do primeiro quadro 40a da camada base e das três primeiras porções de dados 42da camada de otimização. O segundo quadro 44b a ser decodificado com uma taxa de quadro superior exige a informação completa do segundo quadro 40b da camada base e das porções de dados 42b da camada de otimização.

[0022] Um decodificador convencional combinaria todas as unidades NAL das camadas base e de otimização que apresentam a mesma estampa de tempo de decodificação DTS ou estampa de tempo de apresentação PTS. O tempo de remoção da unidade de acesso gerada AU do buffer elementar seria fornecido pela DTS da camada mais elevada (o segundo fluxo de

dados). Porém, a associação de acordo com os valores DTS ou PTS dentro das diferentes camadas não é mais possível, uma vez que os valores dos pacotes de dados correspondentes são diferentes. Para manter a associação de acordo com os valores PTS ou DTS, ainda viável, ao segundo quadro 40b da camada base deveria ser teoricamente atribuído um valor de estampa de tempo de decodificação conforme indicado pelo quadro hipotético 40c da camada base. Porém, um decodificador compatível com o padrão de camada base apenas (o modelo HRD correspondente à camada base) não poderia mais decodificar mesmo a camada base, já que os buffers associados também são pequenos ou a força de processamento também é lenta para decodificar os dois quadros subsequentes com o deslocamento de tempo de decodificação reduzido.

[0023] Em outras palavras, tecnologias convencionais inviabilizam o uso flexível da informação de uma unidade NAL anterior (quadro 40b) em uma camada mais baixa na qualidade de um quadro de referência para informação de decodificação de uma camada mais alta. Porém, essa flexibilidade pode ser necessária, especialmente quando do transporte de vídeo com diferentes taxas de quadro que apresentam razões desiguais como em diferentes camadas de um fluxo SVC. Um importante exemplo pode ser por exemplo um fluxo de vídeo escalável que apresenta uma taxa de quadro de 24 quadros/segundo (conforme usado nas produções cinematográficas) na camada de otimização e 20 quadros/segundo na camada base. Em um cenário desse, isso pode representar uma economia extrema de bits para codificar o primeiro quadro da camada de otimização como um

quadro-p dependendo de um quadro-i 0 da camada base. Os quadros dessas duas camadas apresentariam obviamente porém diferentes estampas de tempo. De-multiplexação apropriada e reordenação para prover uma sequência de quadros na ordem correta para um decodificador subsequente não seria possível com o uso de técnicas convencionais e com mecanismos de fluxo de transporte existentes descritos nos parágrafos anteriores. Visto que ambas contém informação de sincronização diferente para taxas de quadro diferentes, o padrão de fluxo MPEG e outros conhecidos mecanismos de transporte de fluxo de bits para o transporte de fluxos de vídeo escaláveis ou de dados interdependentes não possibilitam a flexibilidade necessária que permite definir ou referenciar as unidades NAL correspondentes ou porções de dados das mesmas imagens em uma camada diferente.

[0024] Existe a necessidade de prover um esquema de referência mais flexível entre porções de dados diferentes de diferentes sub-fluxos contendo porções de dados interrelacionadas.

[0025] De acordo com algumas concretizações da presente invenção, essa possibilidade é viabilizada através de métodos para derivar uma decodificação ou estratégia de associação para porções de dados pertencentes a um primeiro e segundo fluxo de dados em um fluxo de transporte. Os diferentes fluxos de dados contém diferentes informações de sincronização, sendo que as informações de sincronização são definidas de forma que os tempos relativos dentro de um único fluxo de dados sejam consistentes. De acordo com algumas

concretizações da presente invenção, a associação entre porções de dados de diferentes fluxos de dados é obtida mediante inclusão da informação de associação em um segundo fluxo de dados, que precisa referenciar porções de dados de um primeiro fluxo de dados. De acordo com algumas concretizações, a informação de associação referencia um dos campos de dados já existentes dos pacotes de dados do primeiro fluxo de dados. Assim, pacotes individuais no primeiro fluxo de dados podem ser distintamente referenciados através de pacotes de dados do segundo fluxo de dados.

[0026] De acordo com outras concretizações da presente invenção, a informação das primeiras porções de dados referenciadas pelas porções de dados do segundo fluxo de dados e a informação de sincronização das porções de dados no primeiro fluxo de dados. De acordo com outras concretizações, são referenciadas outras informações distintas das primeiras porções de dados do primeiro fluxo de dados, tais como por exemplo, números ID de pacote contínuos, ou similar.

[0027] De acordo com outras concretizações da presente invenção, nenhum dado adicional é introduzido na porções de dados do segundo fluxo de dados enquanto campos de dados já existente são utilizados diferentemente com a finalidade de incluir a informação de associação. Ou seja, por exemplo, campos de dados reservados para informação de sincronização no Segundo fluxo de dados podem ser utilizados para inserir a informação de associação adicional que permite uma referência distinta para porções de dados de diferentes

fluxos de dados.

[0028] Em termos gerais, algumas concretizações da invenção também oferecem a possibilidade de gerar uma representação de dados de vídeo compreendendo um primeiro e um Segundo fluxo de dados nos quais pode ser realizada uma referência flexível entre as porções de dados dos diferentes fluxos de dados no fluxo de transporte.

[0029] Várias concretizações da presente invenção serão a seguir descritas sob referência às figuras inseridas, onde:

[0030] A Fig. 1 mostra um exemplo de de-multiplexação de fluxo de transporte;

[0031] A Fig. 2 mostra um exemplo de de-multiplexação de fluxo de transporte SVC;

[0032] A Fig. 3 mostra um exemplo de um fluxo de transporte SVC;

[0033] A Fig. 4 mostra uma concretização de um método para gerar uma representação de um fluxo de transporte;

[0034] A Fig. 5 mostra uma outra concretização de um método para gerar uma representação de um fluxo de transporte;

[0035] A Fig. 6a mostra uma concretização de um método para derivar uma estratégia de decodificação;

[0036] A Fig. 6b mostra outra concretização de um método para derivar uma estratégia de decodificação

[0037] A Fig. 7 mostra um exemplo de uma sintaxe de fluxo de transporte;

[0038] A Fig. 8 mostra outro exemplo de uma sintaxe de fluxo de transporte;

[0039] A Fig. 9 mostra uma concretização de um gerador de estratégia de decodificação; e

[0040] A Fig. 10 mostra uma concretização de um escalonador de pacote de dados.

[0041] A Fig. 4 descreve uma possível implementação de um método inventivo para gerar uma representação de uma sequência de vídeo em um fluxo de dados de transporte 100. Um primeiro fluxo de dados 102 que apresenta primeiras porções de dados de 102a a 102c e um Segundo fluxo de dados 104 que apresenta as segundas porções de dados de 104a e 104b, são combinados para gerar o fluxo de dados de transporte 100. A informação de associação é gerada a qual associa uma primeira porção de dados pré-determinada do primeiro fluxo de dados 102^a uma segunda porção de dados 106 do segundo fluxo de dados. No exemplo da figura 4, a associação é feita por incorporação da informação de associação 108 na segunda porção de dados 104a. Na concretização ilustrada na figura 4, a informação de associação 108 referencia a primeira informação de sincronização 112 da primeira porção de dados 102a, por exemplo, incluindo um indicador ou copiando a informação de sincronização como a informação de associação. Isso sem falar que outras concretizações podem utilizar outra informação de associação, tais como, por exemplo, números ID de cabeçalho único, números de quadro de fluxo MPEG ou similar.

[0042] Um fluxo de transporte, que compreende a primeira porção de dados 102a e a segunda porção de dados

106a pode portanto ser gerado pela multiplexação das porções de dados na ordem de sua informação de sincronização original.

[0043] Ao invés de introduzir a informação de associação como novos campos de dados exigindo espaço adicional de bit, campos de dados já existentes, tais como, por exemplo, o campo de dados contendo a segunda informação de sincronização 110, podem ser utilizados para receber a informação de associação.

[0044] A Fig. 5 resume uma concretização de um método para gerar uma representação de uma sequência de vídeo que apresenta um primeiro fluxo de dados compreendendo as primeiras porções de dados, sendo que as primeiras porções de dados apresentam a primeira informação de sincronização e o Segundo fluxo de dados compreendendo as segundas porções de dados, sendo que as segundas porções de dados apresentam a segunda informação de sincronização. Em uma etapa de associação 120, a informação de associação é associada a uma segunda porção de dados do Segundo fluxo de dados, sendo que a informação de associação indica uma primeira porção de dados pré-determinada do primeiro fluxo de dados.

[0045] No lado do decodificador, pode ser derivada uma estratégia de decodificação para o fluxo de transporte gerado 210, conforme ilustrado na Fig. 6a. A Fig. 6a ilustra o conceito geral da derivação de uma estratégia de decodificação para uma segunda porção de dados 200 em função de uma porção de dados de referência 402, sendo que a segunda porção de dados 200 é parte de um Segundo fluxo de dados de

um fluxo de transporte 210, sendo que o fluxo de transporte compreende um primeiro fluxo de dados e um segundo fluxo de dados, sendo que a primeira porção de dados 202 do primeiro fluxo de dados compreende a primeira informação de sincronização 212 e a segunda porção de dados 200 do segundo fluxo de dados compreender a segunda informação de sincronização 214 assim como a informação de associação 216 indicando uma primeira porção de dados pré-determinada 202 do primeiro fluxo de dados. Em particular, a informação de associação compreende a primeira informação de sincronização 212 ou uma referência ou indicador para a primeira informação de sincronização 212, permitindo assim identificar de forma distinta a primeira porção de dados 202 no primeiro fluxo de dados.

[0046] A estratégia de decodificação para a segunda porção de dados 200 é derivada utilizando-se a segunda informação de sincronização 214 como sendo a indicação para um tempo de processamento (o tempo de decodificação ou o tempo de apresentação) para a segunda porção de dados e a primeira porção de dados referenciada 202 do primeiro fluxo de dados como uma porção de dados de referência. Ou seja, assim que a estratégia de decodificação é derivada em uma etapa de geração de estratégia 220, as porções de dados podem ser além disso processadas ou decodificadas (no caso de dados de vídeo) através de um método de decodificação subsequente 230. Quando a segunda informação de sincronização 214 é usada como uma indicação para o tempo de processamento t_2 e quando a porção de dados de referência é conhecida, o decodificador

pode ser provido de porções de dados na ordem correta no tempo certo. Ou seja, o conteúdo de dados correspondente à primeira porção de dados 202 é previsto para o decodificador primeiro, seguido pelo conteúdo de dados correspondente à segunda porção de dados 200. O instante de tempo no qual conteúdos de dados são previstos para o decodificador 232 é indicado pela segunda informação de sincronização 214 da segunda porção de dados 200.

[0047] Uma vez que a estratégia de decodificação é derivada, a primeira porção de dados pode ser processada antes da segunda porção de dados. Processamento pode significar em uma concretização que a primeira porção de dados é acessada antes da segunda porção de dados. Em uma outra concretização, o acesso pode compreender a extração de uma informação necessária para decodificar a segunda porção de dados e, um decodificador subsequente. Isso pode ser por exemplo a informação lateral associada ao fluxo de vídeo.

[0048] Nos parágrafos seguintes, é descrita uma concretização particular através da aplicação do conceito inventivo de referenciação flexível de porções de dados para o padrão de fluxo de transporte MPEG (ITU-T Rec. H.222.0 | ISO/IEC 13818-1:2007 FPDAM3.2 (extensões SVC), Antalya, Turquia, Janeiro 2008: [3] ITU-T Rec. H.264 200X 4a edição (SVC) | ISO/IEC 14496-10:200X 4a edição (SVC)).

[0049] Conforme anteriormente resumido, concretizações da presente invenção podem conter, ou adicionar informação adicional para identificar estampas de tempo nos sub-fluxos (fluxos de dados) com valores DID mais

baixos (por exemplo, o primeiro fluxo de dados de um fluxo de transporte compreendendo dois fluxos de dados). A estampa de tempo da unidade de acesso reordenada $A(j)$ é fornecida pelo sub-fluxo com o valor de DID mais alto (o segundo fluxo de dados) ou com o DID máximo quando mais de dois fluxos de dados estão presentes. Embora as estampas de tempo do sub-fluxo com o DID máximo da camada de sistema possam ser usadas para decodificação e/ou sincronização de saída, pode-se obter uma reordenação através da informação adicional de sincronização by additional timing information tref que indica a representação de dependência correspondente no sub-fluxo com outro valor (p.ex o próximo inferior) de DID. Esse procedimento é ilustrado na figura 7. Em algumas concretizações, a informação adicional pode ser transportada para um campo de dados adicional, p.ex para o delimitador de representação de dependência SVC ou, por exemplo, como uma extensão no cabeçalho PES. Alternativamente, ela pode ser transportada para campos de informação de sincronização existentes (p.ex. para os campos de cabeçalho PES), se estiver adicionalmente sinalizado que o conteúdo dos respectivos campos de dados serão alternativamente usados. Na concretização adaptada ao fluxo de transporte MPEG 2 ilustrado na figura 6b, a reordenação pode ser feita conforme abaixo detalhado. A Fig. 6b mostra múltiplas estruturas cujas funcionalidades são descritas através das seguintes abreviações:

[0050] $A_n(j)$ = jth unidade de acesso de fluxo de sub-bits n é decodificada em $tdn(jn)$, onde $n=0$ indica a camada

base

[0051] DIDn = NAL cabeçalho de unidade element
de sintaxe dependência_id em fluxo de sub-bits n

[0052] DPBn = buffer de imagem decodificada de
fluxo de sub-bits

[0053] DRn(jn)= jnth representação de dependência no
fluxo de sub-bits n

[0054] DRBn = buffer de representação de
dependência de fluxo de sub-bits n

[0055] EBn = buffer de fluxo elementar de fluxo de
sub-bits n

[0056] MBn = buffer de multiplexação de fluxo de
sub-bits n

[0057] PIDn = programa ID de fluxo de sub-bits n
no fluxo de transporte.

[0058] TBn = buffer de transporte de fluxo de sub-
bits n

[0059] tdn(jn)= estampa de tempo de decodificação da
representação de dependência jnth no fluxo de sub-bits n

[0060] tdn(jn) pode diferir de pelo menos um tdm(jm)
na mesma unidade de acesso An(j)

[0061] tpn(jn)= apresentação estampa de tempo da
representação de dependência jnth no fluxo de sub-bits n

[0062] tpn(jn) pode diferir de pelo menos um tpm(jm)
na mesma unidade de acesso An(j)

[0063] trefn(Jn)= referência de estampa de tempo ao
fluxo de sub-bits inferior (diretamente referenciado) do jnth

[0064] Representação de dependência no fluxo de sub-

bits n , onde t_{ref} é $t_{refn}(jn)$

[0065] transportada adicionalmente para $tdn(jn)$ está no pacote PES p.ex no delimitador de representação de dependência SVC NAL

[0066] O fluxo de transporte recebido 300 é processado conforme segue.

[0067] Todas as representações de dependência $DRz(jz)$ que iniciam com o valor máximo, $z = n$, na ordem de recebimento jn de $DRn(jn)$ no sub-fluxo sub-stream n . Ou seja, os sub-fluxos são de-multiplexados pelo de-multiplexador 4, conforme indicado pelos números individuais PID. O conteúdo das porções de dados recebido é armazenado nas DRBs das cadeias de buffer individuais dos diferentes fluxos de sub-bits. Os dados das DRBs são extraídos na ordem de z para criar a unidade de acesso $jnth$ $An(jn)$ do sub-fluxo n , de acordo com a seguinte regra:

[0068] Para tanto, considera-se que o fluxo de sub-bits y seja um fluxo de sub-bits que apresenta um DID superior ao do fluxo de sub-bits x . Ou seja, a informação no fluxo de sub-bits y depende da informação no fluxo de sub-bits x . Para cada dois $DRx(jx)$ e $DRy(jy)$ correspondentes, $t_{refy}(jy)$ deve ser igual a $tdx(jx)$. Ao se aplicar esta instrução ao padrão de fluxo de transporte MPEG 2, pode-se obter o seguinte:

[0069] A informação de associação t_{ref} pode ser indicada adicionando-se um campo na extensão de cabeçalho PES, que também pode ser usada através de padrões futuros de codificação escaláveis/multi-view. Para o respectivo campo a

ser analisado, tanto o flag de extensão PES como o flag 2 de extensão PES podem ser ajustados a unidade e o flaf de extensão id de fluxo pode ser ajustado a 0. A informação de associação tref poderia ser sinalizada com o uso do bit reservado da seção de extensão PES.

[0070] Também existe a possibilidade de definir um tipo de extensão adicional PES, que também seria preparado para extensões futuras.

[0071] De acordo com uma outra concretização, um campo de dados adicional para a informação de associação pode ser adicionado ao delimitador de representação de dependência SVC. Assim, um bit de sinalização pode ser introduzido para indicar a presença do novo campo dentro da representação de dependência SVC. Tal bit adicional pode, por exemplo, ser introduzido no descritor SVC ou no descritor de hierarquia.

[0072] De acordo com uma concretização, pode ser implementada extensão do pacote de cabeçalho PES utilizando-se os flags existentes a seguir ou introduzindo-se os seguintes flags adicionais:

[0073]

[0074] Flag de referência de estampa de tempo -

Este é um flag 1-bit, quando ajustado a '1' indicando a presença de.

[0075] Flag de referência PTS_DTS - Este é um flag 1-bit.

[0076] Flags PTR_DTR_- Este é um campo 2-bit. Quando o campo de flags PTR_DTR_flags estiver ajustado a '10', os campos PTR seguintes conterão uma referência a um campo PTS

em outro fluxo de sub-bits SVC de vídeo ou a camada base AVC com o valor próximo inferior de dependência ID de elemento de sintaxe de cabeçalho de unidade NAL conforme presente no fluxo de sub-bits de vídeo SVC que contém esta extensão dentro do cabeçalho PES. Quando o campo de flags PTR_DTR_é ajustado a '01' os campos DTR a seguir conterão uma referência a um campo a DTS em outro fluxo de sub-bits de vídeo SVC ou a camada base AVC com o valor próximo inferior da dependência ID de elemento de sintaxe do cabeçalho de unidade NAL conforme presente no fluxo de sub-bits de vídeo SVC que contém esta extensão dentro do cabeçalho PES. Quando o campo de flags PTR_DTR_ é ajustado a '00' não estarão presentes referências PTS ou DTS no cabeçalho de pacote PES. O valor '11' é proibido.

[0077] PTR (referência do tempo de representação)-
Este é um número 33-bit codificado em três campos separados. Este é uma referência a um campo PTS em outro fluxo de sub-bits de vídeo SVC ou a camada base AVC com o valor próximo inferior de dependência ID de elemento de sintaxe de cabeçalho de unidade NAL conforme presente no fluxo de sub-bits de vídeo SVC contendo esta extensa dentro do cabeçalho PES.

[0078] DTR (referência de tempo de apresentação)
Este é um número 33-bit codificado em três campos separados. Este é uma referência a um campo DTS em outro fluxo de sub-bits de vídeo SVC ou a camada base AVC com o valor próximo inferior de dependência ID de elemento de sintaxe de cabeçalho de unidade NAL conforme presente no fluxo de sub-

bits de vídeo SVC que contém esta extensão dentro do cabeçalho PES.

[0079] Na figura 7 aparece indicado um exemplo de uma sintaxe correspondente que utiliza os flags existentes e outros flags de dados adicionais.

[0080] Um exemplo de uma sintaxe, que pode ser usada ao se implementar a segunda opção anteriormente descrita, aparece ilustrado na figura 8. Para implementar a informação de associação adicional, os elementos de sintaxe a seguir podem ser atribuídos aos seguintes números ou valores:

[0081] Semântica de unidade NAL de delimitador de representação de dependência SVC.

[0082] zero-bit proibido -deve ser igual a 0x00

[0083] nal_ref_idc -deve ser igual a 0x00

[0084] nal_unit_type -deve ser igual a 0x18

[0085] t_ref[32...0] - deve ser igual à estampa de tempo de decodificação DTS conforme indicada no cabeçalho PES para a representação de dependência com o valor próximo inferior de dependência ID de elemento de sintaxe de cabeçalho de unidade NAL da mesma unidade de acesso em um fluxo de sub bits de video SVC ou a camada base AVC. Onde o t_ref é ajustado como segue com relação aos DTS da representação de dependência referenciada: DTS[14..0] é igual a t_ref[14..0], DTS[29..15] é igual a t_ref[29..15], DTS[32..30] é igual a t_ref[32..30].

[0086] Maker_bit - é um campo de 1-bit e deve ser igual a "1".

[0087] Outras concretizações da presente invenção

podem ser implementadas como hardware dedicado ou em circuito integrado de hardware.

[0088] A Fig. 9, por exemplo, mostra um gerador de estratégia de decodificação para uma segunda porção de dados em função de uma porção de dados de referência, sendo que a segunda porção de dados é parte de um segundo fluxo de dados de um fluxo de transporte que compreende um primeiro fluxo e um Segundo fluxo de dados, sendo que as primeiras porções de dados do primeiro fluxo de dados compreende primeira informação de sincronização e sendo que a segunda porção de dados do segundo fluxo de dados compreende segunda informação de sincronização assim como informação de associação que indica primeira porção de dados pré-determinada do primeiro fluxo de dados.

[0089] O gerador de estratégia de decodificação 400 compreende um gerador de informação de referência 402 assim como um gerador de estratégia 404. O gerador de informação de referência 402 é adaptado para derivar a porção de dados de referência para a segunda porção de dados utilizando a primeira porção de dados pré-determinada do primeiro fluxo de dados. O gerador de estratégia 404 é adaptado para derivar a estratégia de decodificação para a segunda porção de dados utilizando a segunda informação de sincronização conforme a indicação para o tempo de processamento para a segunda porção de dados e para a porção de dados de referência derivada pelo gerador de informação de referência 402.

[0090] De acordo com uma outra concretização da presente invenção, um decodificador de vídeo inclui um

gerador de estratégia de decodificação conforme ilustrado na figura 9 a fim de criar uma estratégia de ordem de decodificação para porções de dados de vídeo presente dentro de pacotes de dados de diferentes fluxos de dados associados a diferentes níveis de um codec de vídeo escalável.

[0091] As concretizações da presente invenção, portanto, permitem criar um fluxo de vídeo codificado de modo eficiente, que compreende informação sobre diferentes qualidades de um fluxo de vídeo codificado. Devido à referenciação flexível, uma quantidade significativa de taxa de bit pode ser preservada já que é possível evitar transmissão redundante de informação dentro de camadas individuais.

[0092] A aplicação da referenciação flexível entre diferentes porções de dados de diferentes fluxos de dados não é somente útil no contexto de codificação de vídeo. Em geral, ela pode ser aplicada a qualquer tipo de pacotes de dados de diferentes fluxos de dados.

[0093] A Fig. 10 mostra uma concretização de um programador de pacote de dados 500 compreendendo um gerador de ordem de processo 502, um receptor opcional 504 e um reordenador opcional 506. O receptor é adaptado para receber um fluxo de transporte que compreende um primeiro fluxo de dados e um Segundo fluxo de dados que apresenta uma primeira e segunda porções de dados, sendo que a primeira porção de dados compreende primeira informação de sincronização e sendo que a segunda porção de dados compreende segunda informação de sincronização e informação de associação.

[0094] O gerador de ordem de processo 502 é adaptado para gerar um programa de processamento que apresenta uma ordem de processamento, de tal forma que a segunda porção de dados é processada após a primeira porção de dados referenciada do primeiro fluxo de dados. O reordenador 506 é adaptado para gerar a segunda porção de dados 452 após a primeira porção de dados 450.

[0095] Conforme além disso ilustrado na Fig.10, o primeiro e segundo fluxo de dados não devem estar necessariamente contidos dentro de um fluxo de dados de transporte multiplexado, conforme indicado como opção A. Ao contrário, também é possível transmitir o primeiro e segundo fluxo de dados como fluxos de dados separados, conforme indicado pela opção B da Fig. 10.

[0096] Múltipla transmissão e roteiros de fluxo de dados podem ser melhorados através da referência flexível introduzida nos parágrafos anteriores. Outros roteiros de aplicação são apresentados nos parágrafos seguintes.

[0097] Um fluxo de mídia, com escalável ou multi-view ou multi descrição, ou qualquer outra propriedade, que permita dividir a mídia em subconjuntos lógicos, é transferido através de diferentes canais ou armazenado em diferentes recipientes de armazenamento. A divisão de fluxo de mídia também pode exigir a divisão de frames de mídia individuais ou unidade de acesso necessários em sua totalidade para decodificação em subpartes. Para recuperar a ordem de decodificação dos frames ou unidades de acesso após transmissão através de diferentes canais ou armazenamento em

diferentes recipientes de armazenamento, é necessário um processo para recuperação de ordem de decodificação, já que a confiança na ordem de transmissão nos diferentes canais ou na ordem de armazenamento em diferentes recipientes de armazenamento não pode permitir a recuperação da ordem de decodificação do fluxo completo de mídia ou qualquer subconjunto independentemente aplicável do fluxo completo de mídia. Um sub conjunto A do fluxo de mídia completo é foi estendido de subpartes particulares de unidades de acesso para novas unidades de acesso do subconjunto de fluxo de mídia. Os subconjuntos de fluxo de mídia podem requerer estampas de tempo de decodificação e de apresentação por frame/unidade de acesso dependendo do número de subconjuntos do fluxo de mídia usado para recuperar unidades de acesso. Alguns canais provêem estampas de tempo de decodificação e/ou de apresentação nos canais, que podem ser usados para recuperar ordem de decodificação. Adicionalmente, canais provêem tipicamente a ordem de decodificação dentro do canal através da transmissão ou ordem de armazenamento ou através de meios adicionais. Para recuperar a ordem de decodificação entre diferentes canais ou os diferentes recipientes de armazenamento é necessária informação adicional. Para pelo menos um canal de transmissão ou recipiente de armazenamento, a ordem de decodificação deve ser derivável através de qualquer meio. Ordem de decodificação dos outros canais é portanto fornecidas pela ordem de decodificação derivável mais os valores que indicam um frame/unidade de acesso ou subpartes destes nos diferentes canais de transmissão ou

recipientes de armazenamento que é derivável para a ordem de decodificação. Ponteiros podem ser estampas de tempo de decodificação ou estampas de tempo de apresentação, mas também podem ser seqüências numerosas que indicam ordem de transmissão ou de armazenagem em um canal particular ou recipiente ou podem ser quaisquer outros indicadores que permitem identificar um frame/unidade de acesso no subconjunto de fluxo de mídia que é derivável para a ordem de decodificação.

[0098] Um fluxo de mídia pode ser dividido em subconjuntos de fluxo de mídia e é transportado através de diferentes canais ou armazenado em diferentes recipientes de armazenamento, ou seja, frames de mídia/unidades de acesso de mídia completas ou subconjunto destes estão presentes nos diferentes canais ou diferentes recipientes de armazenamento. A combinação de sub-partes dos frames/unidades de acesso do fluxo de mídia resulta em subconjuntos decodificáveis do fluxo de mídia.

[0099] Pelo menos em um canal de transmissão ou recipiente de armazenamento, a mídia é transportada ou armazenada em ordem de decodificação ou em pelo menos um canal de transmissão ou recipiente de armazenamento a ordem de decodificação é derivável através de quaisquer outros meios.

[00100] Pelo menos, o canal para o qual a ordem de decodificação pode ser recuperada provê pelo menos um indicador, que pode ser usado para identificar um frame particular/unidade de acesso ou subparte destes. Este

indicador é designado a frames/unidades de acesso ou subpartes destes em pelo menos outro canal ou recipiente, que seja então derivável para a ordem de decodificação.

[00101] Ordem de decodificação de frames/unidades de acesso ou sub partes destes em qualquer outro canal ou recipiente que seja derivável para a ordem de decodificação, é dada por identificadores que permitem encontrar frames/unidades de acesso correspondentes ou subpartes destes no canal ou recipiente que seja derivável para a ordem de decodificação. A ordem de decodificação respectiva é então dada pela ordem de decodificação referenciada no canal, que é derivável para a ordem de decodificação.

[00102] Estampas de tempo de decodificação e/ou de apresentação podem ser usadas como indicadores.

[00103] Podem ser usados como indicadores exclusivamente ou adicionalmente indicadores de visualização de um fluxo de mídia de codificação multi-view.

[00104] Podem ser usados como indicadores exclusivamente ou adicionalmente indicadores que indicam uma divisão de um fluxo de mídia de codificação mulit-view.

[00105] Quando estampas de tempo são usadas como indicador, as estampas de tempo do nível mais elevado são usadas para atualizar as estampas de tempo presentes em subpartes inferiores do frame / unidade de acesso para toda a unidade de acesso.

[00106] Embora as concretizações anteriormente descritas se refiram principalmente a codificação de vídeo e transmissão de vídeo, a referenciação flexível não limitada a

aplicações de vídeo. Pelo contrário, todas as outras aplicações de transmissão divididas em pacotes podem se beneficiar com a aplicação de estratégias de decodificação e estratégias de codificação conforme anteriormente descrito, como por exemplo aplicações de fluxo de áudio que utilizam fluxos de áudio de diferente qualidade ou outras aplicações multi-fluxo.

[00107] E isso sem dizer que a aplicação não depende dos canais de transmissão selecionados. Quaisquer tipos de canais de transmissão podem ser usados, tais como por exemplo transmissão over-the-air, transmissão a cabo, transmissão por fibra, teledifusão via satélite, e similares. Além disso, diferentes fluxos de dados podem ser providos por diferentes canais de transmissão. Por exemplo, o canal base de um fluxo que requer apenas largura da banda limitada pode ser transmitido via rede GSM, enquanto somente aqueles que apresentam pronto para uso um telefone celular UMTS podem ser capazes de receber a camada de implementação que exige uma taxa maior de bit.

[00108] Dependendo de certas exigências de implementação dos métodos inventivos, os métodos inventivos podem ser implementados em hardware ou em software. A implementação pode ser feita utilizando-se um meio de armazenamento digital, particularmente um disco, DVD ou um CD que apresente sinais eletronicamente legíveis ali armazenados, que cooperam com o sistema de computador programável de tal forma que os métodos inventivos sejam realizados. Em geral, a presente invenção é, portanto, um

produto de programa de computador com um código de programa armazenado em um portador legível por máquina, sendo o código de programa operativo para executar os métodos inventivos quando o produto de programa de computador roda em um computador. Em outras palavras, os métodos inventivos são, portanto, um programa de computador que apresenta um código de programa para executar pelo menos um dos métodos inventivos quando o programa de computador roda em um computador.

[00109] Embora o exposto adiante tenha sido particularmente mostrado e descrito com referência a concretizações particulares a esse respeito, o versado na técnica perceberá que é possível realizar várias outras alterações na forma e em detalhes sem que o espírito e escopo desta sejam abandonados. Naturalmente que várias alterações poderão ser feitas relativamente à adaptação a diferentes concretizações sem que os conceitos mais amplos aqui descritos e compreendidos pelas reivindicações a seguir sejam abandonados.

REIVINDICAÇÕES

1. "Método para derivar uma programação de processamento para uma segunda porção de dados dependendo de uma porção de dados de referência, a segunda porção de dados sendo parte de um segundo fluxo de dados de um fluxo de transporte, o fluxo de transporte compreendendo o segundo fluxo de dados e um primeiro fluxo de dados compreendendo o primeiro porções de dados, as primeiras porções de dados compreendendo a primeira informação de temporização e a segunda porção de dados do segundo fluxo de dados compreendendo a segunda informação de temporização e informação de associação indicando uma primeira porção de dados predeterminada do primeiro fluxo de dados, caracterizado por compreender:

derivar a programação de processamento tendo uma ordem de processamento de modo que a segunda porção de dados seja processada após a primeira porção de dados predeterminada do primeiro fluxo de dados; e

usar a segunda informação de tempo como uma indicação para um tempo de processamento para a porção de dados de referência.

2. Método, de acordo com a reivindicação 1, caracterizado pelo fato de que a segunda porção de dados está associada à primeira porção de dados predeterminada usando um carimbo de tempo de apresentação da primeira porção de dados predeterminada como a informação de associação, o carimbo de tempo de apresentação indicando uma apresentação tempo da primeira porção de dados predeterminada dentro da primeira camada do fluxo de dados de vídeo múltiplas visões escalável.

3. Método, de acordo com a reivindicação 2, caracterizado pelo fato de que usa ainda uma informação de visualização indicando uma das diferentes visualizações possíveis dentro do fluxo de dados de vídeo de múltiplas visualizações ou uma informação de partição indicando uma das diferentes partições possíveis de um fluxo de dados de vídeo de múltiplas visualizações do primeiro parte de dados como a informação de associação.

4. Método para derivar uma programação de processamento, de acordo com qualquer uma das reivindicações de 1 a 3, caracterizado pelo fato de que compreende ainda:

receber a primeira e a segunda porções de dados;

e

anexar a segunda parte de dados à primeira parte de dados em um fluxo de bits de saída.

5. Programador de pacote de dados, caracterizado por ser adaptado para gerar uma programação de processamento para uma segunda porção de dados dependendo de uma porção de dados de referência, a segunda porção de dados sendo parte de um segundo fluxo de dados de um fluxo de transporte, o fluxo de transporte compreendendo o segundo fluxo de dados e um primeiro fluxo de dados compreendendo primeiras porções de dados, as primeiras porções de dados compreendendo primeiras informações de tempo e a segunda porção de dados do segundo fluxo de dados compreendendo segunda informação de tempo e informação de associação indicando uma primeira porção de dados predeterminada do primeiro fluxo de dados, compreendendo:

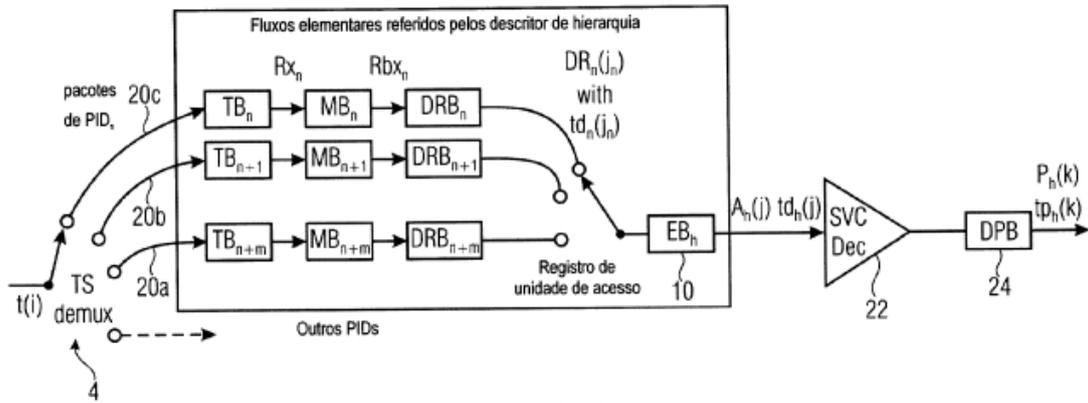
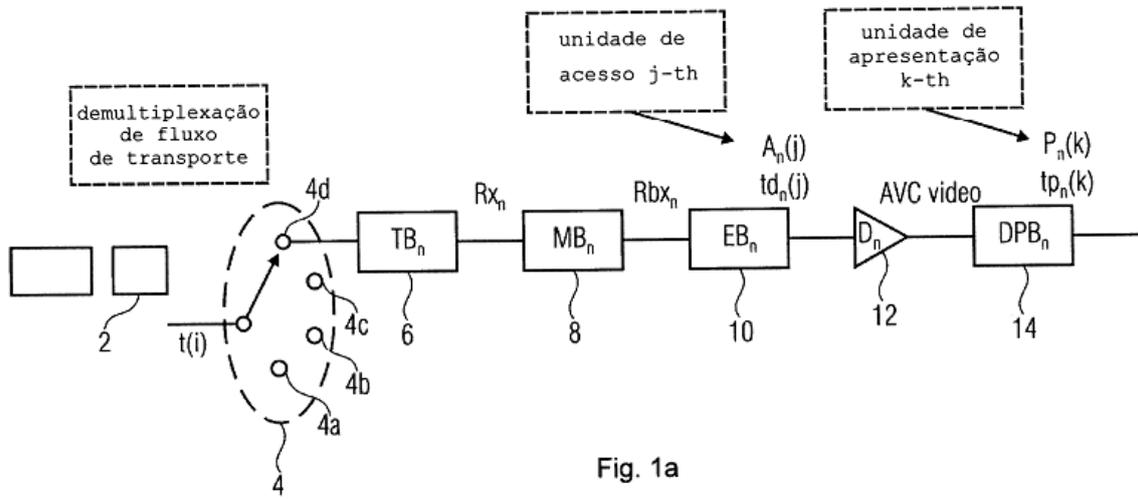
um gerador de ordem de processo adaptado para gerar uma programação de processamento tendo uma ordem de

processamento de modo que a segunda porção de dados seja processada após a primeira porção de dados predeterminada do primeiro fluxo de dados.

6. Programador de pacote de dados, de acordo com a reivindicação 5, caracterizado pelo fato de que compreende ainda:

um receptor adaptado para receber a primeira e a segunda porções de dados; e

um reordenador adaptado para emitir a segunda parte de dados após a primeira parte de dados.



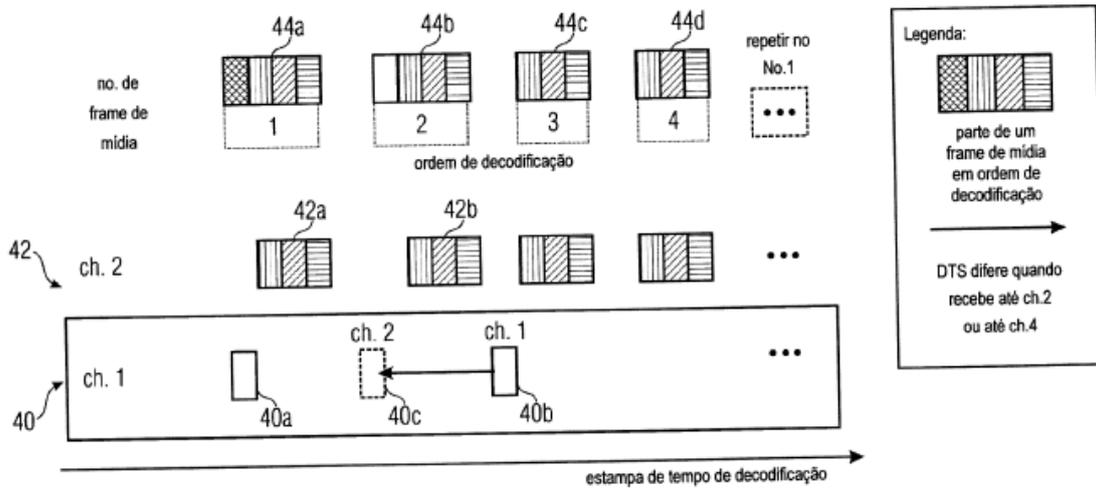


Fig. 3

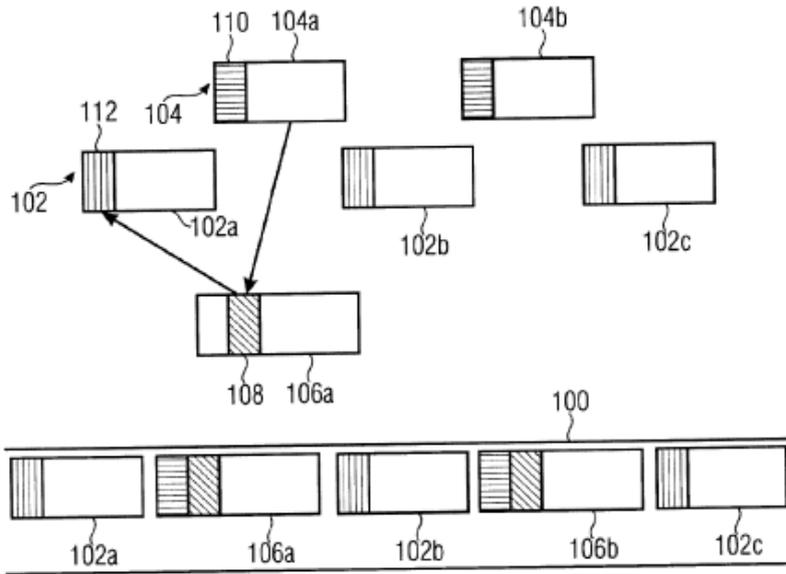


Fig. 4

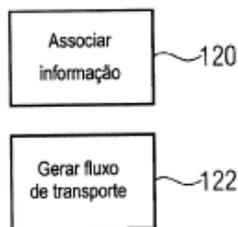


Fig. 5

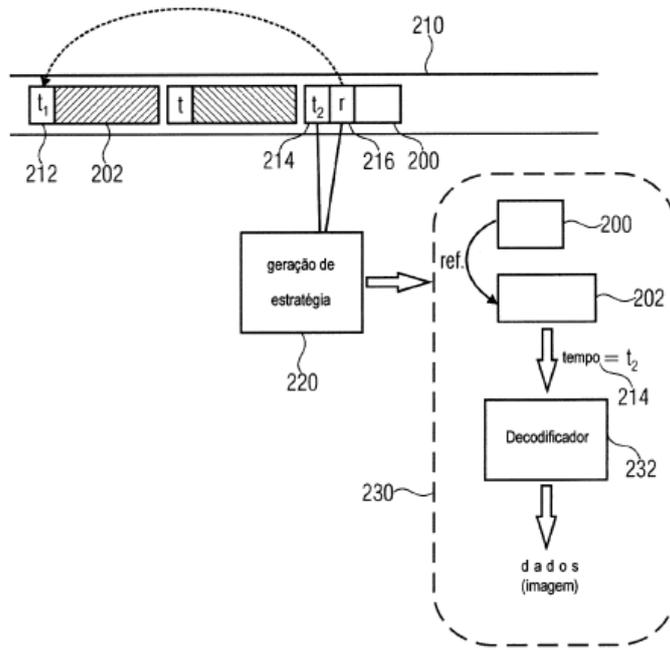


Fig. 6a

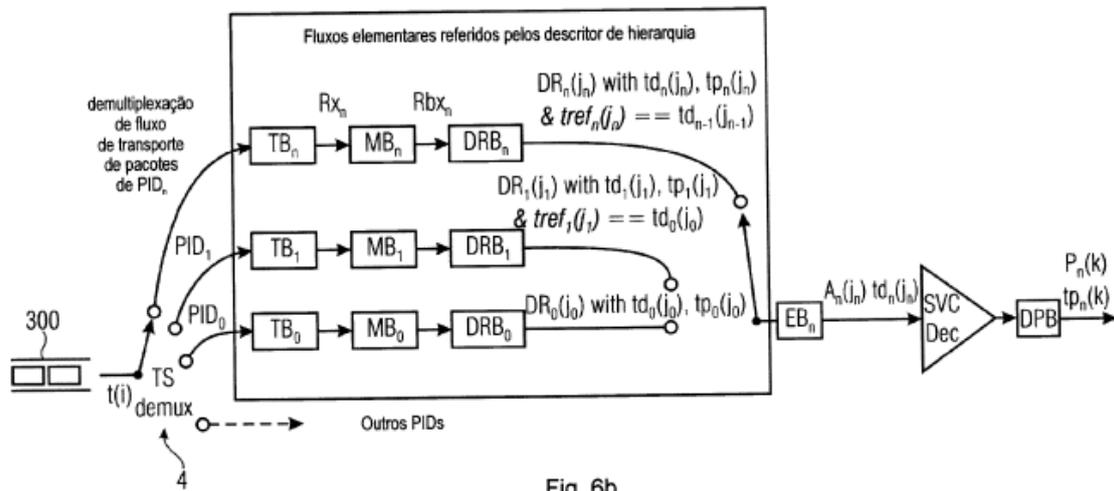


Fig. 6b

| Sintaxe | No. de bits | Mnemônico |
|---------------------------------------|-------------|-----------|
| if (PES_extension_flag_2='1') { | | |
| marker_bit | 1 | bslbf |
| PES_extension_field_length | 7 | uimsbf |
| stream id extension flag | 1 | bslbf |
| if (stream_id_extension_flag = '0') { | | |
| stream id extension | 7 | uimsbf |
| } | | |
| PTR_DTR_flags | 2 | bslbf |
| reserved | 6 | bslbf |
| if (PTR_DTR_flags = '10') { | | |
| reserved | 4 | bslbf |
| PTR [32..30] | 3 | bslbf |
| marker_bit | 1 | bslbf |
| PTR [29..15] | 15 | bslbf |
| marker_bit | 1 | bslbf |
| PTR [14..0] | 15 | bslbf |
| marker_bit | 1 | bslbf |
| } | | |
| if (PTR_DTR_flags='01'){ | | |
| reserved | 4 | bslbf |
| DTR [32..30] | 3 | bslbf |
| marker_bit | 1 | bslbf |
| DTR [29..15] | 15 | bslbf |
| marker_bit | 1 | bslbf |
| DTR [14..0] | 15 | bslbf |
| marker_bit | 1 | bslbf |
| } | | |
| for (i=0; i<N1; i++){ | | |
| reserved | 8 | bslbf |
| } | | |
| } | | |

Fig. 7

| Sintaxe | No. de bits | Mnemônico |
|----------------------|-------------|-----------|
| SVC_drd_nal_unit() { | | |
| forbidden_zero_bit | 1 | bslbf |
| nal_ref_idc | 2 | bslbf |
| nal_unit_type | 5 | bslbf |
| t_ref [32..30] | 3 | bslbf |
| marker_bit | 1 | bslbf |
| t_ref [29..15] | 15 | bslbf |
| marker_bit | 1 | bslbf |
| t_ref [14..0] | 15 | bslbf |
| marker_bit | 1 | bslbf |
| reserved | 24 | uimsbf |
| } | | |

Fig. 8

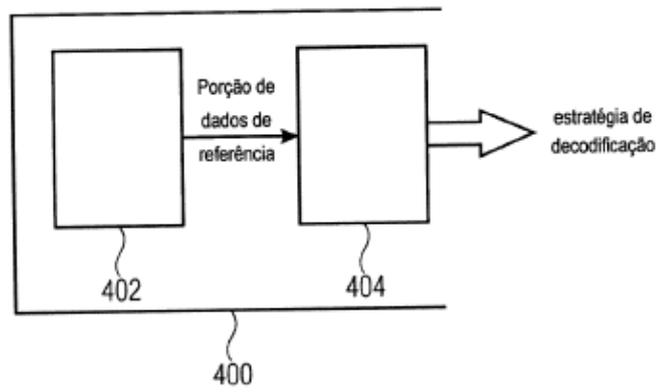


Fig. 9

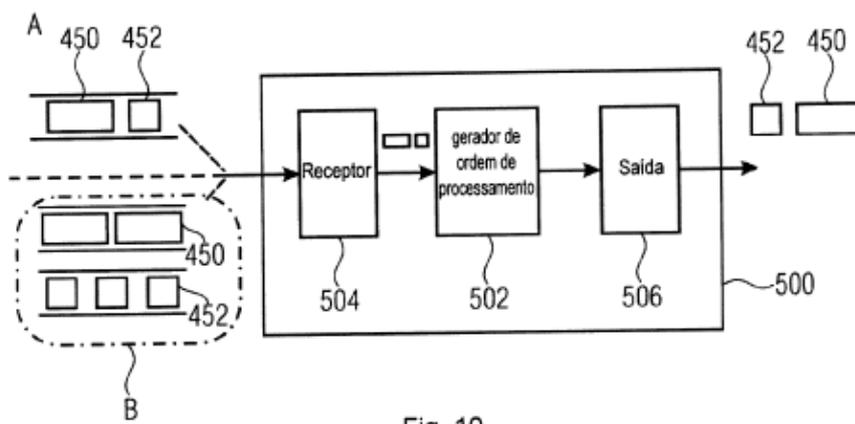


Fig. 10