



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

(21) PI 0906619-5 A2



(22) Data do Depósito: 09/01/2009

(43) Data da Publicação Nacional: 06/08/2009

(54) Título: MÉTODOS PARA CODIFICAR ÁUDIO DE MULTICANAL, PARA CODIFICAR UM FLUXO DE ÁUDIO DE MULTICANAL DE TAXA DE BIT VARIÁVEL SEM PERDA E PARA INICIAR DECODIFICAÇÃO DE UM FLUXO DE BIT DE ÁUDIO DE MULTICANAL DE TAXA DE BIT VARIÁVEL, UM OU MAIS MEIOS LEGÍVEIS POR COMPUTADOR, UM OU MAIS DISPOSITIVOS SEMICONDUTORES, E, DECODIFICADOR DE ÁUDIO DE MULTICANAL

(51) Int. Cl.: G06F 15/00.

(30) Prioridade Unionista: 30/01/2008 US 12/011899.

(71) Depositante(es): DTS, INC.

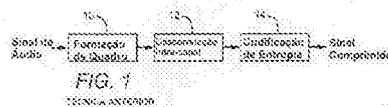
(72) Inventor(es): ZORAN FEJZO.

(86) Pedido PCT: PCT US2009000124 de 09/01/2009

(87) Publicação PCT: WO 2009/097076 de 06/08/2009

(85) Data da Fase Nacional: 29/07/2010

(57) Resumo: MÉTODOS PARA CODIFICAR ÁUDIO DE MULTICANAL, PARA DECODIFICAR UM FLUXO DE ÁUDIO DE MULTICANAL DE TAXA DE BIT VARIÁVEL SEM PERDA E PARA INICIAR DECODIFICAÇÃO DE UM FLUXO DE BIT DE ÁUDIO DE MULTICANAL DE TAXA DEBIT VARIÁVEL, UM OU MAIS MEIOS LEGÍVEIS POR COMPUTADOR, UM ou MAIS DISPOSITIVOS SEMICONDUTORES, E, DECODIFICADOR DE ÁUDIO DE MULTICANAL Um codec de áudio sem perda codifica/decodifica um fluxo de bits de taxa de bit variável (VBR) sem perda com capacidade de ponto de acesso aleatório (RAP) para iniciar decodificação sem perda a um segmento especificado dentro de um quadro e/ou capacidade de múltiplos conjuntos de parâmetro de predição (MPPS) dividido para diminuir efeitos transientes. Isto é realizado com uma técnica de segmentação adaptável que fixa pontos de começo de segmento baseado em constrangimentos impostos pela existência de um RAP desejado e/ou transiente detectado no quadro e seleciona uma duração ótima de segmento em cada quadro para reduzir carga útil de quadro codificado sujeito a um constrangimento de carga útil de segmento codificado RAP e MPPS são particularmente aplicáveis para melhorar desempenho global para durações de quadro mais longas.



“MÉTODOS PARA CODIFICAR ÁUDIO DE MULTICANAL, PARA
 DECODIFICAR UM FLUXO DE ÁUDIO DE MULTICANAL DE TAXA
 DE BIT VARIÁVEL SEM PERDA E PARA INICIAR DECODIFICAÇÃO
 DE UM FLUXO DE BIT DE ÁUDIO DE MULTICANAL DE TAXA DE
 5 BIT VARIÁVEL, UM OU MAIS MEIOS LEGÍVEIS POR COMPUTADOR,
 UM OU MAIS DISPOSITIVOS SEMICONDUTORES, E,
 DECODIFICADOR DE ÁUDIO DE MULTICANAL”

**REFERÊNCIA CRUZADA A PEDIDOS
 RELACIONADOS**

10 Este pedido reivindica benefício de prioridade sob 35 U. S. C.
 120 como uma continuação em parte (CIP) do Pedido US No. 10/911.067
 intitulado "Lossless Multi-Channel Audio Codec", depositado em 4 de agosto
 de 2004, os conteúdos inteiros de qual estão incorporados por referência.

FUNDAMENTO DA INVENÇÃO

Campo da Invenção

15 Esta invenção relaciona-se a codecs de áudio sem perda, e
 mais especificamente a um codec de áudio de multicanal sem perda usando
 segmentação adaptável com capacidade de ponto de acesso aleatório (RAP) e
 capacidade de múltiplo conjunto de parâmetro de predição (MPPS).

Descrição da Arte Relacionada

20 Vários sistemas de codificação de áudio com perda de baixa de
 taxa de bit estão atualmente em uso em uma ampla gama de produtos de
 consumidor e de reprodução de áudio profissionais e serviços. Por exemplo,
 sistema de codificação de áudio Dolby AC3 (Dolby digital) é um padrão
 25 mundial para codificar estéreo e trilhas de som de áudio de 5.1 canais para
 Disco a Laser, vídeo de DVD codificado para NTSC, e ATV, usando taxas de
 bit até 640 kbit/s. Padrões de codificação de áudio de MPEG I e MPEG II são
 usados extensamente para codificação de trilha de som estéreo e multicanal
 para vídeo de DVD codificado para PAL, radiodifusão de rádio digital

terrestre na Europa e Radiodifusão de Satélite nos EUA, a taxas de bit até 768 kbit/s. sistema de codificação de áudio de Acústica Coerente de DTS (Sistemas Digitais de Cinema) é freqüentemente usado para trilhas de som de áudio de 5.1 canais de qualidade de estúdio para Disco Compacto, vídeo de DVD, Radiodifusão de Satélite na Europa e Disco a Laser e taxas de bit até 1536 kbit/s.

Recentemente, muitos consumidores mostraram interesse nestes denominados codecs "sem perda". Codecs "sem perda" se confiam em algoritmos que comprimem dados sem descartar qualquer informação e produzem um sinal decodificado que é idêntico ao sinal de fonte (digitalizado). Este desempenho vem a um custo: tais codecs tipicamente requerem mais largura de banda do que codecs com perda, e comprimem os dados a um grau menor.

Figura 1 é uma representação de diagrama de bloco das operações envolvidas em comprimir sem perda um único canal de áudio. Embora os canais em áudio de multicanal não sejam geralmente independentes, a dependência é freqüentemente fraca e difícil para levar em conta. Portanto, os canais são tipicamente comprimidos separadamente. Porém, alguns codificadores tentarão remover correlação formando um sinal residual simples e codificando (CH1, CH1-CH2). Abordagens mais sofisticadas levam, por exemplo, várias etapas sucessivas de projeção ortogonal através da dimensão de canal. Todas as técnicas estão baseadas no princípio de primeiro remover redundância do sinal e então codificar o sinal resultante com um esquema de codificação digital eficiente. Codecs Sem Perda incluem MPL (Áudio de DVD), áudio Monkey (aplicativos de computador), Apple sem perda, Windows Media Professional sem perda, AudioPak, DVD, LTAC, MUSICcompress, OggSquish, Philips, Shorten, Sonarc e WA. Uma revisão de muitos deste codecs é provida por Mat Hans, Ronald Schafer "Lossless Compression of Digital Audio", Hewlett Packard,

1999.

Formação de quadros 10 é introduzida para prover capacidade de edição, o volume puro de dados proíbe descompressão repetitiva do sinal inteiro precedendo a região a ser editada. O sinal de áudio é dividido em quadros independentes de duração de tempo igual. Esta duração não deveria ser curta demais, desde que custo indireto significativo pode resultar do cabeçalho que é prefixado a cada quadro. Reciprocamente, a duração de quadro não deveria ser longa demais, desde que isto limitaria a capacidade de adaptação temporal e faria edição mais difícil. Em muitas aplicações, o tamanho de quadro é constrangido pela taxa de bit de pico da mídia na qual o áudio é transferido, a capacidade de armazenamento temporário do decodificador e intenção de desejo para ter cada quadro independentemente decodificável.

Descorrelação intra-canal 12 remove redundância descorrelatando as amostras de áudio em cada canal dentro de um quadro. A maioria dos algoritmos remove redundância por algum tipo de modelagem preditiva linear do sinal. Nesta abordagem, um preditor linear é aplicado às amostras de áudio em cada quadro resultando em uma seqüência de amostras de erro de predição. Uma segunda abordagem, menos comum, é obter uma representação quantizada de baixa taxa de bit ou com perda do sinal, e então comprimir sem perda a diferença entre a versão com perda e a versão original. Codificação de entropia 14 remove redundância do erro do sinal residual sem perder qualquer informação. Métodos típicos incluem codificação de Huffman, codificação de comprimento corrido e codificação de Rice. A saída é um sinal comprimido que pode ser reconstruído sem perda.

A especificação de DVD existente e a especificação de HD DVD preliminar fixaram um limite rígido no tamanho de uma unidade de acesso de dados, que representa uma parte do fluxo de áudio, que uma vez extraída pode ser decodificada completamente e as amostras de áudio

reconstruídas enviadas às memórias temporárias de saída. O que isto significa para um fluxo sem perda é que a duração de tempo que cada unidade de acesso pode representar tem que ser pequeno bastante que o pior caso de taxa de bit de pico, a carga útil codificada não excede o limite rígido. A duração de tempo também deve ser reduzida para taxas de amostragem aumentadas e número aumentado de canais, que aumentam a taxa de bit de pico.

Para assegurar compatibilidade, estes codificadores existentes terão que fixar a duração de um quadro inteiro para ser curta bastante para não exceder o limite rígido em uma configuração de frequência de canal/amostragem/largura de bit de pior caso. Na maioria das configurações, isto será fatal e pode degradar seriamente desempenho de compressão. Além disso, esta abordagem de pior caso não gradua bem com canais adicionais.

SUMÁRIO DA INVENÇÃO

A presente invenção provê um codec de áudio que gera um fluxo de bits taxa de bit variável (VBR) sem perda com capacidade de ponto de acesso aleatório (RAP) para iniciar decodificação sem perda a um segmento especificado dentro de um quadro e/ou capacidade de múltiplo conjunto de parâmetro de predição (MPPS) dividido para diminuir efeitos transientes.

Isto é realizado com uma técnica de segmentação adaptável que determina pontos de começo de segmento para assegurar constrangimentos de limite em segmentos impostos pela existência de um RAP desejado e/ou um ou mais transientes no quadro e seleciona uma duração ótima de segmento em cada quadro para reduzir carga útil de quadro codificado sujeito a um constrangimento de carga útil de segmento codificado. Em geral, os constrangimentos de limite especificam que um RAP desejado ou transiente deve se achar dentro de um certo número de blocos de análise do começo de um segmento. Em uma concretização exemplar na qual segmentos dentro de um quadro são da mesma duração e uma potência de

dois da duração de bloco de análise, uma duração máxima de segmento é determinada para assegurar que as condições desejadas sejam satisfeitas. RAP e MPPS são particularmente aplicáveis para melhorar desempenho global para durações de quadro mais longas.

5 Em uma concretização exemplar, um fluxo de bits de áudio VBR sem perda é codificado com RAPs (segmentos de RAP) alinhados dentro de uma tolerância especificada de RAPs desejados providos em um código de temporização de codificador. Cada quadro é bloqueado em uma seqüência de blocos de análise com cada segmento tendo uma duração igual
10 àquela de um ou mais blocos de análise. Em cada quadro sucessivo até um bloco de análise de RAP é determinado do código de temporização. O local do bloco de análise de RAP e um constrangimento que o bloco de análise de RAP deve se achar dentro de M blocos de análise do começo do segmento de RAP fixa um começo de um segmento de RAP. Parâmetros de predição são
15 determinados para o quadro, dois conjuntos de parâmetros (por canal) se MPPS estiver habilitado e um transiente é detectado em um canal. As amostras no quadro de áudio são comprimidas com a predição sendo desabilitada para as primeiras amostras até a ordem de predição seguindo o começo do segmento de RAP. Segmentação adaptável é empregada nas
20 amostras residuais para determinar uma duração de segmento e parâmetros de codificação de entropia para cada segmento para minimizar a carga útil de quadro codificado sujeito ao começo fixado do segmento de RAP e os constrangimentos de carga útil de segmento codificado. Parâmetros de RAP indicando a existência e local do segmento de RAP e dados de navegação são
25 empacotados no cabeçalho. Em resposta a um comando de navegação para iniciar reprodução tal como seleção de usuário de uma cena ou observação, o decodificador desempacota o cabeçalho do próximo quadro no fluxo de bits para ler os parâmetros de RAP até que um quadro incluindo um segmento de RAP seja detectado. O decodificador extrai duração de segmento e dados de

navegação para navegar ao começo do segmento de RAP. O decodificador desabilita predição para as primeiras amostras até que uma história de predição seja reconstruída e então decodifica o resto dos segmentos e quadros subsequentes em ordem, desabilitando o preditor cada vez que um segmento de RAP é encontrado. Esta construção permite a um decodificador iniciar decodificação a ou muito perto de RAPs especificados por codificador com uma resolução de sub-quadro. Isto é particularmente útil com durações de quadro mais longas ao tentar sincronizar reprodução de áudio a um código de temporização de vídeo que especifica RAPs, por exemplo, ao começo de capítulos.

Em outra concretização exemplar, um fluxo de bits de áudio VBR sem perda é codificado com MPPSSs divididos de forma que transientes detectados estejam localizados dentro dos primeiros L blocos de análise de um segmento nos seus canais respectivos. Em cada quadro sucessivo até um transiente por canal por conjunto de canal e seu local dentro do quadro é detectado. Parâmetros de predição são determinados para cada partição considerando os pontos começo de segmento impostos pelos transientes. As amostras em cada partição são comprimidas com o conjunto de parâmetro respectivo. Segmentação adaptável é empregada nas amostras residuais para determinar uma duração de segmento e parâmetros de codificação de entropia para cada segmento para minimizar a carga útil de quadro codificado sujeito aos constrangimentos de começo de segmento impostos pelos transientes (e RAP) e os constrangimentos de carga útil de segmento codificados. Parâmetros transientes indicando a existência e local do primeiro segmento transiente (por canal) e dados de navegação são empacotados no cabeçalho. Um decodificador desempacota o cabeçalho de quadro para extrair os parâmetros transientes e conjunto adicional de parâmetros de predição. Para cada canal em um conjunto de canal, o decodificador usa o primeiro conjunto de parâmetros de predição até que o segmento transiente seja encontrado e

troca ao segundo conjunto para o resto do segmento. Embora a segmentação do quadro seja a mesma por canais e múltiplos conjuntos de canal, o local de um transiente (se qualquer) pode variar entre conjuntos e dentro de conjuntos. Esta construção permite a um decodificador trocar conjuntos de parâmetro de predição a ou muito perto do início de transientes detectados com uma resolução de sub-quadro. Isto é particularmente útil com durações de quadro mais longas para melhorar eficiência de codificação global.

Desempenho de compressão pode ser ademais aumentado formando $M/2$ canais de descorrelação para áudio de M canais. A trinca de canais (base, correlatado, descorrelatado) provê duas possíveis combinações de par (base, correlatado) e (base, descorrelatado), que pode ser considerado durante a segmentação e otimização de codificação de entropia para ademais melhorar desempenho de compressão. Os pares de canal podem ser especificados por segmento ou por quadro. Em uma concretização exemplar, o codificador forma os dados de áudio e então extrai pares de canal ordenados incluindo um canal de base e um canal correlatado e gera um canal descorrelatado para formar pelo menos uma trinca (base, correlatado, descorrelatado). Se o número de canais for ímpar, um canal de base extra é processado. Predição polinomial adaptável ou fixa é aplicada a cada canal para formar sinais residuais. Para cada trinca, o par de canal (base, correlatado) ou (base, descorrelatado) com a carga útil codificada menor é selecionado. Usando o par de canal selecionado, um conjunto global de parâmetros de codificação pode ser determinado para cada segmento através de todos os canais. O codificador seleciona o conjunto global ou conjuntos distintos de parâmetros de codificação baseado em qual tem a carga útil codificada total menor (cabeçalho e dados de áudio).

Em qualquer abordagem, uma vez que o conjunto ótimo de parâmetros de codificação e pares de canal para a partição atual (duração de segmento) tenham sido determinados, o codificador calcula a carga útil

codificada em cada segmento por todos os canais. Assumindo que os
constrangimentos em começo de segmento e tamanho máximo de carga útil
de segmento para quaisquer RAPs desejados ou transientes detectados são
satisfeitos, o codificador determina se a carga útil codificada total para o
quadro inteiro para a partição atual é menos que a corrente ótima para uma
partição anterior. Se verdadeiro, o conjunto atual de parâmetros de
codificação e carga útil codificada são armazenados e a duração de segmento
é aumentada. O algoritmo de segmentação começa apropriadamente dividindo
o quadro nos tamanhos de segmento mínimos igual ao tamanho de bloco de
análise e aumenta a duração de segmento por uma potência de dois a cada
etapa. Este processo se repete até que tanto o tamanho de segmento viole o
constrangimento de tamanho máximo ou a duração de segmento cresça à
duração máxima de segmento. A habilitação das características de RAP ou
MPPS e a existência de um RAP desejado ou transiente detectado dentro de
um quadro pode fazer a rotina de segmentação adaptável escolher uma
duração de segmento menor que caso contrário faria.

Estas e outras características e vantagens da invenção serão
aparentes àqueles qualificados na arte da descrição detalhada seguinte de
concretizações preferidas, tomadas junto com os desenhos acompanhantes,
em que:

DESCRIÇÃO BREVE DOS DESENHOS

Figura 1, como descrita acima, é um diagrama de bloco para
um codificador de áudio sem perda padrão;

Figuras 2a e 2b são diagramas de bloco de um codificador e
decodificador de áudio sem perda, respectivamente, conforme a presente
invenção;

Figura 3 é um diagrama de informação de cabeçalho como
relacionado à segmentação e seleção de código de entropia;

Figuras 4a e 4b são diagramas de bloco do processamento de

janela de análise e processamento de janela de análise inversa;

Figura 5 é um fluxograma de descorrelação de canal cruzado;

Figuras 6a e 6b são diagramas de bloco de análise de predição adaptável e processamento e processamento de predição adaptável inversa;

5 Figuras 7a e 7b são um fluxograma de segmentação ótima e seleção de código de entropia;

Figuras 8a e 8b são fluxogramas de seleção de código de entropia para um conjunto de canal;

10 Figura 9 é um diagrama de bloco de um codec de extensão de núcleo mais sem perda;

Figura 10 é um diagrama de um quadro de fluxo de bit no qual cada quadro inclui um cabeçalho e uma pluralidade de segmentos;

Figuras 11a e 11b são diagramas de informação de cabeçalho adicional relacionada à especificação de RAPs e MPPSSs;

15 Figura 12 é um fluxograma para determinar limites de segmento ou uma duração máxima de segmento para RAPs desejados ou transientes detectados:

Figura 13 é um fluxograma para determinar MPPSSs;

20 Figura 14 é um diagrama de um quadro ilustrando a seleção de pontos de começo de segmento ou uma duração máxima de segmento;

Figuras 15a e 15b são diagramas ilustrando o fluxo de bits e decodificação do fluxo de bits a um segmento de RAP e um transiente; e

25 Figura 16 é um diagrama ilustrando segmentação adaptável baseada nos constrangimentos de carga máxima útil de segmento e duração máxima de segmento.

DESCRIÇÃO DETALHADA DA INVENÇÃO

A presente invenção provê um algoritmo de segmentação adaptável que gera um fluxo de bits de taxa de bit variável (VBR) sem perda com capacidade de ponto de acesso aleatório (RAP) para iniciar decodificação

sem perda a um segmento especificado dentro de um quadro e/ou capacidade de múltiplos conjuntos de parâmetro de predição (MPPS) dividido para diminuir efeitos transientes. A técnica de segmentação adaptável determina e fixa pontos de começo de segmento para assegurar que condições de limite impostas por RAPs desejados e/ou transientes detectados sejam satisfeitas e seleciona uma duração ótima de segmento em cada quadro para reduzir carga útil de quadro codificado sujeito a um constrangimento de carga útil de segmento codificado e os pontos começo de segmento fixos. Em geral, os constrangimentos de limite especificam que um RAP desejado ou transiente deve se achar dentro de um certo número de blocos de análise do começo de um segmento. O RAP desejado pode ser mais ou menos o número de blocos de análise do começo de segmento. O transiente se acha dentro do primeiro número de blocos de análise do segmento. Em uma concretização exemplar na qual segmentos dentro de um quadro são da mesma duração e uma potência de dois da duração de análise de bloco, uma duração máxima de segmento é determinada para assegurar as condições desejadas. RAP e MPPS são particularmente aplicáveis para melhorar desempenho global para durações de quadro mais longas.

CODEC DE ÁUDIO SEM PERDA

Como mostrado nas Figuras 2a e 2b, os blocos operacionais essenciais são semelhantes a codificadores e decodificadores sem perda existentes com a exceção de modificações para o processamento de janelas de análise para fixar condições de começo de segmento para RAPs e/ou transientes e a segmentação e seleção de código de entropia. Um processador de janelas de análise sujeita o áudio de PCM de multicanal a processamento de janela de análise, que bloqueia os dados em quadros de uma duração constante, fixa pontos começo de segmento baseado em RAPs desejados e/ou transientes detectados e remove redundância descorrelatando as amostras de áudio em cada canal dentro de um quadro. Descorrelação é

executada usando predição, que é definida amplamente para ser qualquer processo que usa amostras de áudio reconstruídas antigas (a história de predição) para estimar um valor para uma amostra original atual e determinar um resíduo. Técnicas de predição abrangem fixa ou adaptável e linear ou não linear, entre outras. Em vez de codificar em entropia os sinais residuais diretamente, um segmentador adaptável executa um processo de segmentação ótima e seleção de código de entropia 24 que segmenta os dados em uma pluralidade de segmentos e determina a duração de segmento e parâmetros de codificação, por exemplo, a seleção de um codificador de entropia particular e seus parâmetros, para cada segmento que minimiza a carga útil codificada para o quadro inteiro sujeito ao constrangimento que cada segmento deve ser decodificável completamente e sem perda, menos que um número máximo de bytes menos que o tamanho de quadro, menos que a duração de quadro, e que qualquer RAP desejado e/ou transiente detectado deve se achar dentro de um número especificado de blocos de análise (resolução de sub-quadro) do começo de um segmento. Os conjuntos de parâmetros de codificação são otimizados para cada canal distinto e podem ser otimizados para um conjunto global de parâmetros de codificação. Um codificador de entropia codifica em entropia 26 cada segmento de acordo com seu conjunto particular de parâmetros de codificação. Um empacotador empacota 28 dados codificados e informação de cabeçalho em fluxo de bits 30.

Como mostrado na Figura 2b, para executar a operação de decodificação, o decodificador navega a um ponto no fluxo de bits 30 em resposta a, por exemplo, seleção de usuário de uma cena de vídeo ou capítulo ou passeio de usuário, e um desempacotador desempacota o fluxo de bits 40 para extrair a informação de cabeçalho e dados codificados. O decodificador desempacota informação de cabeçalho para determinar o próximo segmento de RAP ao qual decodificação pode começar. O decodificador então navega ao segmento de RAP e inicia decodificação. O decodificador desabilita

predição para um certo número de amostras quando encontra cada segmento de RAP. Se o decodificador detectar a presença de transiente em um quadro, o decodificador usa um primeiro conjunto de parâmetros de predição para decodificar uma primeira partição e então usa um segundo conjunto de parâmetros de predição para decodificar do transiente adiante dentro do quadro. Um decodificador de entropia executa uma decodificação de entropia 42 em cada segmento de cada canal de acordo com os parâmetros de codificação nomeados para reconstruir sem perda os sinais residuais. Um processador de janelas de análise inversa sujeita estes sinais a processamento de janela de análise inversa 44, que executa predição inversa para reconstruir sem perda o áudio de PCM original 20.

NAVEGAÇÃO DE FLUXO DE BITS E FORMATO DE CABEÇALHO

Como mostrado na Figura 10, um quadro 500 em fluxo de bits 30 inclui um cabeçalho 502 e uma pluralidade de segmentos 504. Cabeçalho 502 inclui um 'sync' 506, um cabeçalho comum 508, um sub-cabeçalho 510 para o um ou mais conjuntos de canal, e dados de navegação 512. Nesta concretização, dados de navegação 512 incluem um cepo de NAVI 514 e código de correção de erros CRC16 516. O cepo de NAVI preferivelmente fragmenta os dados de navegação nas porções menores do fluxo de bits para habilitar navegação completa. O cepo inclui segmentos de NAVI 518 para cada segmento e cada segmento de NAVI inclui um tamanho de carga útil de NAVI Ch Set 520 para cada conjunto de canal. Entre outras coisas, isto permite ao decodificador navegar ao começo do segmento de RAP para qualquer conjunto de canal especificado. Cada segmento 504 inclui os resíduos codificados em entropia 522 (e amostras originais onde predição desabilitada para RAP) para cada canal em cada conjunto de canal.

O fluxo de bits inclui informação de cabeçalho e dados codificados para pelo menos um e preferivelmente múltiplos conjuntos de

canal diferentes. Por exemplo, um primeiro conjunto de canal pode ser uma configuração 2.0, um segundo conjunto de canal pode ser um adicional 4 canais constituindo uma apresentação de 5.1 canais, e um terceiro conjunto de canal pode ser um 2 canais de som circundante adicionais constituindo apresentação de 7.1 canais global. Um decodificador de 8 canais extrairia e decodificaria todos os 3 conjuntos de canal produzindo uma apresentação de 7.1 canais a suas saídas. Um decodificador de 6 canais extrairá e decodificará conjunto de canal 1 e conjunto de canal 2 ignorando completamente o conjunto de canal 3 produzindo a apresentação de 5.1 canais. Um decodificador de 2 canais só extrairá e decodificará conjunto de canal 1 e ignorará conjuntos de canal 2 e 3 produzindo uma apresentação de 2 canais. Tendo o fluxo estruturado desta maneira permite capacidade de expansão de complexidade de decodificador.

Durante a codificação, um codificador de tempo executa a denominada "mistura abaixo embutida" tal que mistura abaixo de 7.1 \rightarrow 5.1 esteja prontamente disponível em 5.1 canais que são codificados em conjuntos de canal 1 e 2. Semelhantemente, uma mistura abaixo de 5.1 \rightarrow 2.0 está prontamente disponível em 2.0 canais que são codificados como um conjunto de canal 1. Um decodificador de 6 canais decodificando conjuntos de canal 1 e 2 obterá mistura abaixo de 5.1 depois de desfazer a operação de embutir mistura abaixo de 5.1 \rightarrow 2.0 executada no lado de codificação. Semelhantemente, um decodificador de 8 canais completo obterá apresentação de 7.1 original decodificando conjuntos de canal 1, 2 e 3 e desfazendo a operação de embutir mistura abaixo de 7.1 \rightarrow 5.1 e 5.1 \rightarrow 2.0 executada no lado de codificação.

Como mostrado na Figura 3, o cabeçalho 32 inclui informação adicional além do que é provida ordinariamente para um codec sem perda a fim de implementar a segmentação e seleção de código de entropia. Mais especificamente, o cabeçalho inclui informação de cabeçalho comum 34 tal

como o número de segmentos (NumSegments) e o número de amostras em cada segmento (NumSamplesInSegm), informação de cabeçalho de conjunto de canal 36 tais como os coeficientes de decorrelação quantizados (QuantChDecorrCoeff [] []) e informação de cabeçalho de segmento 38 tal como o número de bytes em segmento atual para o conjunto de canal (ChSetByteCOns), um indicador de otimização global (AllChSameParamFlag) e indicadores de codificador de entropia (RiceCodeFlag [], CodeParam []), que indicam se codificação de Rice ou Binária é usada e o parâmetro de codificação. Esta configuração de cabeçalho particular assume segmentos de duração igual dentro de um quadro e segmentos que são uma potência de dois da duração de bloco de análise. Segmentação do quadro é uniforme por canais dentro de um conjunto de canal e por conjuntos de canal.

Como mostrado na Figura 11a, o cabeçalho ademais inclui parâmetros de RAP 530 no cabeçalho comum que especificam a existência e local de um RAP dentro de uma dado quadro. Nesta concretização, o cabeçalho inclui uma indicador de RAP = VERDADEIRO se RAP estiver presente. O ID de RAP especifica o número de segmento do segmento de RAP para iniciar decodificação ao acessar o fluxo de bits no RAP desejado. Alternadamente, um RAP_MASK poderia ser usado para indicar segmentos que são e não um RAP. O RAP estará consistente por todos os conjuntos de canal.

Como mostrado na Figura 11b, o cabeçalho inclui: AdPredOrder [0] [ch] = ordem do Preditor Adaptável ou FixedPredOrder [0] [ch] = ordem do Preditor Fixado para canal ch em qualquer quadro inteiro ou no caso de transiente, uma primeira partição do quadro antes de um transiente. Quando predição adaptável é selecionada (AdPredOrder [0] [ch]>0), coeficientes de predição adaptáveis são codificados e empacotados em AdPredCodes[0] [ch] [AdPredOrder [0] [ch]].

No caso de MPPS, o cabeçalho ademais inclui parâmetros transientes 532 na informação de cabeçalho de conjunto de canal. Nesta concretização, cada cabeçalho de conjunto de canal inclui um indicador ExtraPredSetsPrsent [ch] = VERDADEIRO se o transiente for detectado em canal ch, StartSegment [ch] = índice indicando o segmento de começo transiente para canal ch, e AdPredOrder [1] [ch] = ordem do Preditor Adaptável ou FixedPredOrder [1] [ch] = ordem do Preditor Fixo para canal ch aplicável à segunda partição no quadro passado e incluindo um transiente. Quando predição adaptável é seleccionada (AdPredOrder [1] [ch]>0), um segundo conjunto de coeficientes de predição adaptável é codificado e empacotado em AdPredCodes [1] [ch] [AdPredOrder [1] [ch]]. A existência e local de um transiente podem variar pelos canais dentro de um conjunto de canal e por conjuntos de canal.

PROCESSAMENTO DE JANELAS DE ANÁLISE

Como mostrado nas Figuras 4a e 4b, uma concretização exemplar de processamento de janelas de análise 22 selecciona de qualquer predição adaptável 46 ou predição polinomial fixa 48 para descorrelatar cada canal, que é uma abordagem bastante comum. Como será descrito em detalhes com referência à Figura 6a, uma ordem ótima de preditor é estimada para cada canal. Se a ordem for maior que zero, predição adaptável é aplicada. Caso contrário, a predição polinomial fixa mais simples é usada. Semelhantemente, no decodificador, o processamento de janelas de análise inversa 44 selecciona de qualquer predição adaptável inversa 50 ou predição polinomial inversa fixa 52 para reconstruir áudio de PCM dos sinais residuais. As ordens de preditor adaptável e índices de coeficiente de predição adaptável e ordens de preditor fixas são empacotadas 53 na informação de cabeçalho de conjunto de canal.

Descorrelação de Canal Cruzado

Conforme a presente invenção, desempenho de compressão

pode ser ademais aumentado implementando descorrelação de canal cruzado 54, que ordena os M canais de entrada em pares de canal de acordo com uma medida de correlação entre os canais (um "M" diferente do M constrangimento de bloco de análise em um ponto de RAP desejado). Um dos 5 canais é designado como o canal "base" e o outro é designado como o canal "correlatado". Um canal descorrelatado é gerado para cada par de canal para formar um "trinca" (base, correlatado, descorrelatado). A formação da trinca provê duas possíveis combinações de par (base, correlatado) e (base, descorrelatado) que podem ser consideradas durante a segmentação e 10 otimização de codificação de entropia para ademais melhorar desempenho de compressão (veja Figura 8a).

A decisão entre (base, correlatado) e (base, descorrelatado) pode ser executada tanto antes (baseado em alguma medida de energia) ou integrada com segmentação adaptável. A abordagem anterior reduz 15 complexidade enquanto a posterior aumenta eficiência. Uma abordagem 'híbrida' pode ser usada, onde para trincas que têm um canal descorrelatado com variância consideravelmente menor (baseado em um limiar) do que o canal correlatado, uma substituição simples do canal correlatado pelo canal descorrelatado antes de segmentação adaptável é usada enquanto para todas as 20 outras trincas, a decisão sobre codificar canal correlatado ou descorrelatado é deixada ao processo de segmentação adaptável. Isto simplifica a complexidade do processo de segmentação adaptável um pouco sem sacrificar eficiência de codificação.

O PCM de M-ch original 20 e o PCM descorrelatado de M/2- 25 ch 56 são ambos remetidos à predição adaptável e operações de predição polinomiais fixas, que geram sinais residuais para cada um dos canais. Como mostrado na Figura 3, índices (OrigChOrder []) que indicam a ordem original dos canais antes da escolha executada durante o processo de descorrelação em par e um indicador PWChDecorrFlag [] para cada par de canal indicando a

presença de um código para coeficientes de descorrelação quantizados são armazenados no cabeçalho de conjunto de canal 36 na Figura 3.

Como mostrado na Figura 4b, para executar a operação de decodificação de processamento de janela de análise inversa 44, a informação de cabeçalho é desempacotada 58 e os resíduos (amostras originais a começo de segmento de RAP) são passados tanto por predição polinomial fixa inversa 52 ou predição adaptável inversa 50 de acordo com a informação de cabeçalho, isto é as ordens de preditor adaptável e fixa para cada canal. Na presença de um transiente em um canal, o conjunto de canal terá dois conjuntos diferentes de parâmetros de predição para aquele canal. O áudio de PCM descorrelatado de M canais (M/2 canais são descartados durante segmentação) é passado por descorrelação de canal de cruzado inversa 60, que lê os índices de OrigChOrder [] e indicador de PWChDecorriFlag [] do cabeçalho de conjunto de canal e reconstrói sem perda o áudio de PCM de M canais 20.

Um processo exemplar para executar descorrelação de canal cruzado 54 é ilustrado na Figura 5. Por meio de exemplo, o áudio de PCM é provido como M=6 canais distintos, L, R, C, Ls, Rs e LFE, que também corresponde diretamente a uma configuração de conjunto de canal armazenada no quadro. Outros conjuntos de canais podem ser, por exemplo, deixados de envolvente atrás de centro cerque e direito de envolvente atrás de centro para produzir áudio envolvente de 7.1. O processo começa iniciando uma malha de quadro e começando uma malha de conjunto de canal (etapa 70). A estimativa de autocorrelação de atraso zero para cada canal (etapa 72) e a estimativa de correlação cruzada de atraso zero para todas as possíveis combinações de pares de canais no conjunto de canal (etapa 74) são calculadas. A seguir, coeficientes de correlação em par de canal CORCOEF são estimados como a estimativa de correlação cruzada de atraso zero dividida pelo produto das estimativas de autocorrelação de atraso zero para os

canais envolvidos no par (etapa 76). Os CORCOEFs são ordenados do valor absoluto maior ao menor e armazenados em uma tabela (etapa 78). A partir do topo da tabela, índices de par de canal correspondentes são extraídos até que todos os pares tenham sido configurados (etapa 80). Por exemplo, os 6 canais podem ser casados baseado no seu CORCOEF como (L, R), (Ls, Rs) e (C, LFE).

O processo começa uma malha de par de canais (etapa 82), e seleciona um canal "base" como o com a estimativa de autocorrelação de atraso zero menor, que é indicativo de uma energia mais baixa (etapa 84). Neste exemplo, os canais L, Ls e C formam os canais base. O coeficiente de decorrelação de par de canais (ChPairDecorrCoeff) é calculado como a estimativa de correlação cruzada de atraso zero dividida pela estimativa de autocorrelação de atraso zero do canal base (etapa 86). O canal decorrelatado é gerado multiplicando as amostras de canal base com o ChPairDecorrCoeff e subtraindo esse resultado das amostras correspondentes do canal correlatado (etapa 88). Os pares de canais e seu canal decorrelatado associado definem os "trincas" (L, R, R- ChPairDecorrCoeff [1]*L), (Ls, Rs, Rs- ChPairDecorrCoeff [2]*Ls), (C, LFE, LFE-ChPairDecorrCoeff [3]*C) (etapa 89). O ChPairDecorrCoeff [] para cada par de canal (e cada conjunto de canal) e os índices de canal que definem a configuração de par são armazenados na informação de cabeçalho de conjunto de canal (etapa 90). Este processo se repete para cada conjunto de canal em um quadro e então para cada quadro no áudio de PCM em janelas (etapa 92).

Determine Ponto de Começo de Segmento para RAP e Transientes

Uma abordagem exemplar para determinar começo de segmento e constrangimentos de duração para acomodar RAPs desejados e/ou transientes detectados é ilustrada nas Figuras 12 por 14. O bloco mínimo de dados de áudio que é processado é chamado um "bloco de análise". Blocos de

análise só são visíveis no codificador, o decodificador só processa segmentos. Por exemplo, um bloco de análise pode representar 0,5 ms de dados de áudio em um quadro de 32 ms incluindo 64 blocos de análise. Segmentos são incluídos de um ou mais blocos de análise. Idealmente, o quadro é dividido de forma que um RAP desejado ou transiente detectado se ache no primeiro bloco de análise do RAP ou segmento de transiente. Porém, dependendo do local do RAP desejado ou transiente para assegurar esta condição podem forçar uma segmentação subótima (durações de segmento curtas demais) que aumenta carga útil de quadro codificado demais. Portanto, um compromisso é especificar que qualquer RAP desejado deve se achar dentro de M blocos de análise ("M" diferente dos M canais em rotina de decorrelação de canal) do começo do segmento de RAP e qualquer transiente deve se achar dentro dos primeiros L blocos de análise seguindo o começo do segmento de transiente no canal correspondente. M e L são menos que o número total de blocos de análise no quadro e escolhidos para assegurar uma tolerância de alinhamento desejada para cada condição. Por exemplo, se um quadro incluir 64 blocos de análise, M e/ou L poderia ser 1, 2, 4, 8 ou 16. Tipicamente, alguma potência de dois menos que o total e tipicamente uma fração pequena disso (não mais que 25%) para prover resolução de sub-quadro verdadeira. Além disso, embora duração de segmento possa ser permitida variar dentro de um quadro para fazer assim complica grandemente o algoritmo de segmentação adaptável e aumenta bits de cabeçalho com uma melhoria relativamente pequena em eficiência de codificação. Conseqüentemente, uma concretização típica constrange os segmentos para serem de duração igual dentro de um quadro e de uma duração igual a uma potência de dois da duração de bloco de análise, por exemplo duração de segmento = 2^P * duração de bloco de análise, onde $P = 0, 1, 2, 4, 8$, etc. No caso mais geral, o algoritmo especifica o começo dos segmentos de RAP ou transientes. No caso constrangido, o algoritmo especifica uma duração máxima de segmento para cada quadro que

assegura que as condições sejam satisfeitas.

Como mostrado na Figura 12, um código de temporização de codificação incluindo RAPs desejados tal como um código de temporização vídeo que especifica começos de capítulo ou cena é provido pela camada de aplicativo (etapa 600). Tolerâncias de alinhamento que ditam os valores max de M e L acima são providas (etapa 602). Os quadros são formados em uma pluralidade de blocos de análise e sincronizados ao código de temporização para alinhar RAPs desejados a blocos de análise (etapa 603). Se um RAP desejado se achar dentro do quadro, o codificador fixa o começo de um segmento de RAP onde o bloco de análise de RAP deve se achar dentro de M blocos de análise antes ou depois do começo do segmento de RAP (etapa 604). Note, o RAP desejado pode se achar de fato no segmento precedendo o segmento de RAP dentro de M blocos de análise do começo do segmento de RAP. A abordagem começa a análise de Predição Adaptável/Fixa (etapa 605), começa a Malha de Conjunto de Canal (etapa 606) e começa a Análise de Predição Adaptável/Fixa no conjunto de canal (etapa 608) chamando a rotina ilustrada na Figura 13. A Malha de Conjunto de Canal termina (etapa 610) com a rotina retornando o um conjunto de parâmetros de predição (AdPredOrder[0] [], FixedPredOrder [0] [] e AdPredCodes [0] [] []) para o caso quando ExtraPredSetsPresent [] = FALSO ou dois conjuntos de parâmetros de predição (AdPredOrder [0] [], FixedPredOrder [0] [], AdPredCodes [0] [] [], AdPredOrder [1] [], FixedPredOrder [1] [] e AdPredCodes [1] [] []) para o caso quando ExtraPredSetsPresent [] = VERDADEIRO, os resíduos e o local de qualquer transiente detectado (StartSegment []) por canal (etapa 612). Etapa 608 é repetida para cada conjunto de canal que é codificado no fluxo de bits. Pontos de começo de segmento para cada quadro são determinados do ponto de começo de segmento de RAP e/ou pontos de começo de segmento transientes detectados e passados ao algoritmo de segmentação adaptável das Figuras 16 e 7a-7b

(etapa 614). Se as durações de segmento forem constrangidas para serem uniformes e uma potência de dois do comprimento de bloco de análise, uma duração máxima de segmento é selecionada baseado nos pontos começo fixados e passado ao algoritmo de segmentação adaptável (etapa 616). O constrangimento de duração máxima de segmento mantém a vantagem de pontos começo fixos mais adicionando um constrangimento em duração.

Uma concretização exemplar da Análise de Predição Adaptável/Fixa em uma rotina de Conjunto de Canal (etapa 608) é provida na Figura 13. A rotina começa malha de canal indexada por ch (etapa 700), computa coeficientes de predição baseados em quadro e coeficientes de predição baseados em partição (se um transiente for detectado) e seleciona a abordagem com a melhor eficiência de codificação por canal. É possível que até mesmo se um transiente for detectado, a codificação mais eficiente é ignorar o transiente. A rotina retorna os conjuntos de parâmetro de predição, resíduos e o local de qualquer transiente codificado.

Mais especificamente, a rotina executa uma análise de predição baseada em quadro chamando a rotina de predição adaptável diagramada na Figura 6a (etapa 702) para selecionar um conjunto de parâmetros de predição baseado em quadro (etapa 704). Este único conjunto de parâmetros é então usado para executar predição no quadro de amostras de áudio considerando o começo de qualquer segmento de RAP no quadro (etapa 706). Mais especificamente, predição é desabilitada ao começo do segmento de RAP para as primeiras amostras até a ordem da predição. Uma medida da norma residual baseada em quadro por exemplo a energia residual é estimada dos valores residuais e das amostras originais onde predição está desabilitada.

Em paralelo, a rotina detecta se qualquer transiente existe no sinal original para cada canal dentro do quadro atual (etapa 708). Um limiar é usado para equilibrar entre detecção falsa e detecção perdida. Os índices do bloco de análise contendo um transiente são registrados. Se um transiente for

detectado, a rotina fixa o ponto começo de um segmento de transiente que é posicionado para assegurar que o transiente se ache dentro dos primeiros L blocos de análise do segmento (etapa 709) e divide o quadro em primeira e segunda partições, com a segunda partição coincidente com o começo do segmento de transiente (etapa 710). A rotina então chama a rotina de predição adaptável diagramada na Figura 6a (etapa 712) duas vezes para selecionar primeiro e segundo conjuntos de parâmetros de predição baseado em partição para a primeira e segunda partições (etapa 714). Os dois conjuntos de parâmetros são então usados para executar predição na primeira e segunda partições de amostras de áudio, respectivamente, também considerando o começo de qualquer segmento de RAP no quadro (etapa 716). Uma medida da norma residual baseada em partição (por exemplo energia residual) é estimada dos valores residuais e das amostras originais onde predição está desabilitada.

A rotina compara a norma residual baseada em quadro à norma residual baseada em partição multiplicada por um limiar para responder pela informação de cabeçalho aumentada requerida para múltiplas partições para cada canal (etapa 716). Se a energia residual baseada em quadro for menor, então os resíduos baseados em quadro e parâmetros de predição (etapa 718) são retornados, caso contrário os resíduos baseados em partição, dois conjuntos de parâmetros de predição e o índices dos transientes registrados são retornados para aquele canal (etapa 720). A Malha de Canal indexada por canal (etapa 722) e Análise de Predição Adaptável/Fixa em um conjunto de canal (etapa 724) iteram através dos canais em um conjunto e todos os conjuntos de canal antes de terminar.

A determinação dos pontos de começo de segmento ou duração máxima de segmento para um único quadro 800 é ilustrada na Figura 14. Assuma que quadro 800 é 32 ms e contém 64 blocos de análise 802, cada um 0,5 ms em duração. Um código de temporização de vídeo 804 especifica

um RAP desejado 806 que cai dentro do 9º bloco de análise. Transientes 808 e 810 são detectados em CH 1 e 2 que caem dentro do 5º e 18º blocos de análise, respectivamente. No caso não constrangido, a rotina pode especificar pontos de começo de segmento a blocos de análise 5, 9 e 18 para assegurar que o RAP e transientes se achem no 1º bloco de análise dos seus segmentos respectivos. O algoritmo de segmentação adaptável poderia ademais dividir o quadro para satisfazer outros constrangimentos e minimizar carga útil de quadro contanto que estes pontos de começo sejam mantidos. O algoritmo de segmentação adaptável pode alterar os limites de segmento e ainda pode cumprir a condição que o RAP desejado ou transiente caia dentro de um número especificado de blocos de análise a fim de cumprir outros constrangimentos ou otimizar melhor a carga útil.

No caso constrangido, a rotina determina uma duração máxima de segmento que, neste exemplo, satisfaz as condições em cada um do RAP desejado e os dois transientes. Desde que o RAP desejado 806 cai dentro do 9º bloco de análise, a duração de segmento max assegura que o RAP se acharia no 1º bloco de análise do segmento de RAP é 8x (graduado por duração do bloco de análise). Portanto, os tamanhos de segmento permissíveis (como um múltiplo de dois dos blocos de análise) são 1, 2, 4 e 8. Semelhantemente, desde transiente de Ch 1 808 cai dentro do 5º bloco de análise, a duração máxima de segmento é 4. Transiente 810 em CH 2 é mais problemático visto que assegurar que ele ocorre no primeiro bloco de análise requer uma duração de segmento igual para o bloco de análise (1X). Porém, se o transiente puder ser posicionado no segundo bloco de análise, então a duração de segmento max é 16x. Sob estes constrangimentos, a rotina pode selecionar uma duração de segmento max de 4, por esse meio permitindo ao algoritmo de segmentação adaptável selecionar de 1x, 2x e 4x para minimizar carga útil de quadro e satisfazer os outros constrangimentos.

Em uma concretização alternativa, o primeiro segmento de

todo n-ésimo quadro pode prefixadamente ser um segmento de RAP a menos que o código de temporização especifique um segmento de RAP diferente nesse quadro. O RAP padrão pode ser útil, por exemplo, para permitir a um usuário saltar ao redor ou "surfar" dentro do fluxo de bits de áudio em lugar de ser constrangido a só esses RAPs especificados pelo código de temporização de vídeo.

Predição Adaptável

Análise de Predição Adaptável e Geração Residual

Predição linear tenta remover a correlação entre as amostras de um sinal de áudio. O princípio básico de predição linear é predizer um valor de amostra $s(n)$ usando as amostras prévias $s(n-1)$, $s(n-2)$, ..., e subtrair o valor predito $\hat{s}(n)$ da amostra original $s(n)$. O sinal residual resultante $e(n) = s(n) - \hat{s}(n)$ será idealmente não correlatado e conseqüentemente terá um espectro de freqüência plano. Além disso, o sinal residual terá uma variância menor que o sinal original implicando que menos bits são necessários para sua representação digital.

Em uma concretização exemplar do codec de áudio, um modelo de preditor de FIR é descrito pela equação seguinte:

$$e(n) = s(n) - Q\left\{\sum_{k=1}^M a_k * s(n-k)\right\}$$

onde $Q\{\}$ denota a operação de quantização, M denota a ordem de preditor e a_k são coeficientes de predição quantizados. Uma quantização particular $Q\{\}$ é necessária para compressão sem perda desde que o sinal original é reconstruído no lado de decodificação, usando várias arquiteturas de processador de precisão finita. A definição de $Q\{\}$ está disponível a ambos codificador e decodificador e reconstrução do sinal original é obtida simplesmente por:

$$s(n) = e(n) + Q\left\{\sum_{k=1}^M a_k * s(n-k)\right\}$$

onde é assumido que os mesmos a_k coeficientes de predição quantizados estão disponíveis a ambos codificador e decodificador. Um novo conjunto de parâmetros de preditor é transmitido por cada janela de análise (quadro) permitindo ao preditor adaptar à estrutura sinal de áudio variada em tempo. No caso de detecção de transiente, dois novos conjuntos de parâmetros de predição são transmitidos para o quadro para cada canal no qual um transiente é detectado; um para decodificar resíduos antes do transiente e um para decodificar resíduos que incluindo e subsequente ao transiente.

Os coeficientes de predição são projetados para minimizar o resíduo de predição médio quadrado. A quantização $Q\{\}$ faz o preditor um preditor de não linear. Porém, na concretização exemplar, a quantização é feita com precisão de 24 bits e é razoável assumir que os efeitos não lineares resultantes podem ser ignorados durante otimização de coeficiente de preditor. Ignorando a quantização $Q\{\}$, o problema de otimização subjacente pode ser representado como um conjunto de equações lineares envolvendo os atrasos de seqüência de autocorrelação sinal e os coeficientes de preditor desconhecidos. Este conjunto de equações lineares pode ser resolvido eficientemente usando o algoritmo Levinson-Durbin (LD).

Os coeficientes de predição linear resultantes (LPC) precisam ser quantizados, tal que eles possam ser transmitidos eficientemente em um fluxo codificado. Infelizmente, quantização direta de LPC não é a abordagem mais eficiente desde que os pequenos erros de quantização podem causar grandes erros espectrais. Uma representação alternativa de LPCs é a representação de coeficiente de reflexão (RC), que exhibe menos sensibilidade aos erros de quantização. Esta representação também pode ser obtida do algoritmo de LD. Por definição do algoritmo de LD, os RCs são garantidos terem magnitude ≤ 1 (ignorando erros numéricos). Quando o valor absoluto dos RCs está perto de 1, a sensibilidade de predição linear aos erros de quantização presentes em RCs quantizados fica alta. A solução é executar

quantização não uniforme de RCs com etapas de quantização mais finas ao redor da unidade. Isto pode ser alcançado em duas etapas:

1) transforme RCs a uma representação de relação de log-área (LAR) por meio de função de mapeamento

$$LAR = \log \frac{1+RC}{1-RC}$$

5 onde log denota logaritmo de base natural.

2) quantize uniformemente os LARs.

A transformação de $RC \rightarrow LAR$ deforma a escala de amplitude de parâmetros tal que o resultado de etapas 1 e 2 seja equivalente à quantização não uniforme com etapas de quantização mais finas ao redor da
10 unidade.

Como mostrado na Figura 6a, em uma concretização exemplar de análise de predição adaptável, parâmetros de LAR quantizados são usados para representar parâmetros de preditor adaptáveis e transmitidos no fluxo codificado de bits. Amostras em cada canal de entrada são processadas
15 independente entre si e conseqüentemente a descrição só considerará processamento em um único canal.

A primeira etapa é calcular a seqüência de autocorrelação através da duração de janela de análise (quadro inteiro ou partições antes e depois de um transiente detectado) (etapa 100). Para minimizar os efeitos de
20 bloqueio que são causados por descontinuidades nos dados de limites de quadro é primeiro colocado em janela. A seqüência de autocorrelação para um número especificado (igual a ordem máxima de $LP + 1$) de atrasos é estimada do bloco em janela de dados.

O algoritmo de Levinson-Durbin (LD) é aplicado ao conjunto
25 de atrasos de autocorrelação estimados e o conjunto de coeficientes de reflexão (RC), até a ordem de LP max, é calculado (etapa 102). Um resultado intermediário do algoritmo (LD) é um conjunto de variâncias estimadas de resíduos de predição para cada ordem de predição linear até a ordem de LP

max. No próximo bloco, usando este conjunto de variâncias residuais, a ordem de preditor linear (AdPredOrder) é selecionada (etapa 104).

5 Para a ordem de preditor selecionado, o conjunto de coeficientes de reflexão (RC) é transformado, para o conjunto de parâmetros de relação de log-área (LAR) usando a função de mapeamento declarada acima (etapa 106). Uma limitação do RC é introduzida antes de transformação a fim de prevenir divisão por 0:

$$RC = \begin{cases} Tresh & \forall RC > Tresh \\ -1 & \forall RC < -1 \\ RC & \text{Caso contrário} \end{cases}$$

onde Tresh denota número perto, mas menor que 1.

10 Os parâmetros de LAR são quantizados (etapa 108) de acordo com a regra seguinte:

$$QLARInd = \begin{cases} \left\lfloor \frac{LAR}{q} \right\rfloor & \forall LAR \geq 0 \\ -\left\lfloor \frac{-LAR}{q} \right\rfloor & \forall LAR < 0 \end{cases}$$

onde QLARInd denota os índices de LAR quantizados, $[x]$ indica operação de achar maior valor inteiro menor ou igual a x , e q denota tamanho de etapa de quantização. Na concretização exemplar, região $[-8 \text{ a } 8]$ é codificada usando 8 bits, isto é,

$$q = \frac{2^8}{2}$$

15 e conseqüentemente, QLARInd está limitado de acordo com:

$$QLARInd = \begin{cases} 127 & \forall QLARInd > 127 \\ -127 & \forall QLARInd < -127 \\ QLARInd & \text{Caso contrário} \end{cases}$$

QLARInd são traduzidos de valores assinados a não assinados usando o mapeamento seguinte:

$$AdPredCodes = \begin{cases} 2 * QLARInd & \forall QLARInd \geq 0 \\ 2 * (-QLARInd) - 1 & \forall QLARInd < 0 \end{cases}$$

No bloco "RC LUT", uma quantização inversa de parâmetros de LAR e uma tradução para parâmetros de RC é feita em uma única etapa usando uma tabela de consulta (etapa 112). Tabela de consulta consiste em valores quantizados do mapeamento inverso RC \rightarrow LAR, isto é, mapeamento LAR \rightarrow RC dado por:

$$RC = \frac{e^{LAR} - 1}{e^{LAR} + 1}$$

A tabela de consulta é calculada a valores quantizados de LARs igual a 0, 1,5*q, 2,5*q, ... 127,5*q. Os valores de RC correspondentes, depois de graduar por 2^{16} , são arredondados a inteiros não assinados de 16 bits e armazenados como Q16 números de ponto fixos não assinados em uma tabela de entrada 128.

Parâmetros de RC quantizados são calculados da tabela e os índices de LAR de quantização QLARInd como:

$$QRC = \begin{cases} TABELA[QLARInd] & \forall QLARInd \geq 0 \\ -TABELA[-QLARInd] & \forall QLARInd < 0 \end{cases}$$

Os parâmetros de RC quantizados QRC_{ord} para $ord = 1, \dots, AdPredOrder$ são traduzidos aos parâmetros de predição linear quantizados (LP_{ord} para $ord = 1, \dots, AdPredOrder$) de acordo com o algoritmo seguinte (etapa 114):

For $ord = 0$ to $AdPredOrder - 1$ do

For $m = 1$ to ord do

$$C_{ord+1,m} = C_{ord,m} + (QRC_{ord+1} * C_{ord,ord+1-m} + (1 \ll 15)) \gg 16$$

end

$$C_{ord+1,ord+1} = QRC_{ord+1}$$

end

For $ord = 0$ to $AdPredOrder - 1$ do

$$LP_{ord+1} = C_{AdPredOrder,ord+1}$$

end

Desde que os coeficientes de RC quantizados eram representados em formato de ponto fixo assinado Q16, o algoritmo anterior gerará os coeficientes de LP também em formato de ponto fixo assinado Q16. O caminho de computação de decodificador sem perda é projetado para suportar resultados intermediários até 24 bits. Portanto, é necessário executar uma verificação de saturação depois que cada $C_{ord+1,m}$ é calculado. Se a saturação ocorrer em qualquer fase do algoritmo, o indicador de saturação é fixado e a ordem de preditor adaptável AdPredOrder, para um canal particular, é reajustada a 0 (etapa 116). Para este canal particular com AdPredOrder=0, uma predição de coeficiente fixo será executada em vez da predição adaptável (Veja Predição de Coeficiente Fixo). Note que os índices de quantização de LAR não assinados (PackLARInd [n] para n=1, ..., AdPredOrder [Ch]) são empacotados só no fluxo codificado para os canais com AdPredOrder [Ch] > 0.

Finalmente, para cada canal com AdPredOrder > 0, a predição linear adaptável é executada e os resíduos de predição $e(n)$ são calculados de acordo com as equações seguintes (etapa 118):

$$\overline{s(n)} = \left[\left\{ \sum_{k=1}^{AdPredOrder} LP_k * s(n-k) \right\} + (1 \ll 15) \right] \gg 16$$

$$\text{Limit } \overline{s(n)} \text{ to 24-bit range } (-2^{23} \text{ to } 2^{23} - 1)$$

$$e(n) = s(n) + \overline{s(n)}$$

$$\text{Limit } e(n) \text{ to 24-bit range } (-2^{23} \text{ to } 2^{23} - 1)$$

$$\text{for } n = AdPredOrder + 1, \dots, NumSamples$$

Desde que a meta de projeto na concretização exemplar é que um segmento de RAP específico de certos quadros sejam "pontos de acesso aleatório", a história de amostra não é levada através do segmento precedente para o segmento de RAP. Ao invés, a predição só é comprometida na amostra AdPredOrder+1 no segmento de RAP.

Os resíduos de predição adaptável $e(n)$ são ademais codificados em entropia e empacotados no fluxo codificado de bits.

Predição Adaptável Inversa no Lado de Decodificação

No lado de decodificação, a primeira etapa ao executar predição adaptável inversa é desempacotar a informação de cabeçalho (etapa 120). Se o decodificador estiver tentando iniciar decodificação de acordo com um código de temporização de reprodução (por exemplo seleção de usuário de um capítulo ou surfando), o decodificador acessa o fluxo de bits de áudio próximo, mas antes desse ponto e pesquisa o cabeçalho do próximo quadro até que ache um `RAP_Flag = VERDADEIRO` indicando a existência de um segmento de RAP no quadro. O decodificador então extrai o número de segmento de RAP (ID de RAP) e dados de navegação (NAVI) para navegar ao começo do segmento de RAP, desabilita predição até `índice > pred_order` e inicia decodificação sem perda. O decodificador decodifica os segmentos restantes nos quadros e quadros subseqüentes, desabilitando predição cada vez que um segmento de RAP é encontrado. Se um `ExtraPredSetsPrsnt = VERDADEIRO` for encontrado em um quadro para um canal, o decodificador extrai o primeiro e segundo conjuntos de parâmetros de predição e o segmento de começo para o segundo conjunto.

As ordens de predição adaptável `AdPredOrder [Ch]` para cada canal `Ch=1, ..., NumCh` são extraídas. A seguir, para os canais com `AdPredOrder [Ch] > 0`, a versão não assinada de quantização índices de LAR (`AdPredCodes [n]` para `n=1, ..., AdPredOrder [Ch]`) é extraído. Para cada canal `Ch` com ordem de predição `AdPredOrder [Ch] > 0`, os `AdPredCodes` não assinados `[n]` são mapeados aos valores assinados `QLARInd[n]` usando o mapeamento seguinte:

$$QLARInd[n] = \begin{cases} AdPredCodes[n] \gg 1 & \forall \text{ numeração par } AdPredCodes[n] \\ -(AdPredCodes[n] \gg 1) - 1 & \forall \text{ numeração ímpar } AdPredCodes[n] \end{cases}$$

para $n = 1, \dots, AdPredOrder[Ch]$

onde `>>` denota uma operação de deslocamento à direita de inteiro.

Uma quantização inversa de parâmetros de LAR e uma

tradução para parâmetros de RC é feita em uma única etapa usando um Quant RC LUT (etapa 122). Esta é a mesma tabela de consulta TABLE{} como definida no lado de codificação. Os coeficientes de reflexão quantizados para cada canal Ch [QRC[n] para $n = 1, \dots, \text{AdPredOrder}[\text{Ch}]$] são calculados da TABELA {} e os índices de LAR de quantização QLARInd[n], como:

$$QRC[n] = \begin{cases} TABELA[QLARInd[n]] & \forall QLARInd[n] \geq 0 \\ -TABELA[-QLARInd[n]] & \forall QLARInd[n] < 0 \end{cases}$$

para $n = 1, \dots, \text{PrOr}[\text{Ch}]^{31}$

Para cada canal Ch, os parâmetros de RC quantizados QRC_{ord} para $ord = 1, \dots, \text{AdPredOrder}[\text{Ch}]$ são traduzidos aos parâmetros de predição lineares quantizados (LP_{ord} para $ord = 1, \dots, \text{AdPredOrder}[\text{Ch}]$) de acordo com o algoritmo seguinte (etapa 124):

For $ord = 0$ to $\text{AdPredOrder} - 1$ do

For $m = 1$ to ord do

$$C_{ord+1,m} = C_{ord,m} + (QRC_{ord+1} * C_{ord,ord+1-m} + (1 \ll 15)) \gg 16$$

end

$$C_{ord+1,ord+1} = QRC_{ord+1}$$

end

For $ord = 0$ to $\text{AdPredOrder} - 1$ do

$$LP_{ord+1} = C_{\text{AdPredOrder},ord+1}$$

end

10 Qualquer possibilidade de saturação de resultados intermediários é removida no lado de codificação. Portanto, no lado de codificação não há nenhuma necessidade para executar verificação de saturação depois de cálculo de cada $C_{ord+1,m}$.

15 Finalmente, para cada canal com $\text{AdPredOrder}[\text{Ch}] > 0$, uma predição linear adaptável inversa é executada (etapa 126). Assumindo que resíduos de predição $e(n)$ são extraídos previamente e decodificados em entropia, os sinais originais reconstruídos $s(n)$ são calculados de acordo com as equações seguintes:

$$\overline{s(n)} = \left[\left\{ \sum_{k=1}^{\text{AdPredOrder}[Ch]} LP_k * s(n-k) \right\} + (1 \ll 15) \right] \gg 16$$

Limit $\overline{s(n)}$ to 24-bit range $(-2^{23} \text{ to } 2^{23} - 1)$

$$e(n) = s(n) - \overline{s(n)}$$

para $n = \text{AdPredOrder}[Ch] + 1, \dots, \text{NumSamples}$

Desde que a história de amostra não é mantida a um segmento de RAP, a predição adaptável inversa deverá começar da $(\text{AdPredOrder}[Ch] + 1)$ amostra no segmento de RAP.

Predição de Coeficiente Fixo

5 Uma forma de coeficiente fixo muito simples do preditor linear foi achada ser útil. Os coeficientes de predição fixos são derivados de acordo com um método de aproximação polinomial muito simples proposto primeiro por Shorten (T. Robinson. Shorten: "Simple lossless and near lossless waveform compression", Relatório técnico 156, Departamento de Engenharia da Universidade de Cambridge, Rua Trumpington, Cambridge 10 CB2 1PZ, REINO UNIDO, dezembro de 1994). Neste caso, os coeficientes de predição são aqueles especificados ajustando um polinômio de ordem p aos últimos p pontos de dados. Expandindo em quatro aproximações.

$$\hat{s}_0[n] = 0$$

$$\hat{s}_1[n] = s[n-1]$$

$$\hat{s}_2[n] = 2s[n-1] - s[n-2]$$

$$\hat{s}_3[n] = 3s[n-1] - 3s[n-2] + s[n-3]$$

15 Uma propriedade interessante destas aproximações de polinômios é que o sinal residual resultante, $e_k[n] = s[n] - \hat{s}_k[n]$ pode ser implementado eficientemente da maneira de recursiva seguinte.

$$e_0[n] = s[n]$$

$$e_1[n] = e_0[n] - e_0[n-1]$$

$$e_2[n] = e_1[n] - e_1[n-1]$$

$$e_3[n] = e_2[n] - e_2[n-1]$$

...

A análise de predição de coeficiente fixo é aplicada em uma base por quadro e não se confia em amostras calculadas no quadro prévio ($e_k[-1] = 0$). O conjunto residual com a magnitude de soma menor através de quadro inteiro é definido como a melhor aproximação. A ordem residual
5 ótima é calculada para cada canal separadamente e acumulada no fluxo como Ordem de Predição Fixa (FPO[Ch]). O resíduos $e_{\text{FPO[Ch]}}[n]$ no quadro atual são ademais codificados em entropia e empacotados no fluxo.

O processo de predição de coeficiente fixo inverso, no lado de codificação, é definido por uma fórmula recursiva de ordem k para o cálculo de
10 k -ésima ordem residual a exemplo de amostragem n :

$$e_k[n] = e_{k-1}[n] + e_k[n-1]$$

onde o sinal original desejado $s[n]$ é dado por:

$$s[n] = e_0[n]$$

e onde para cada k -ésimo resíduo de ordem $e_k[-1] = 0$.

Como um exemplo, recursões para a predição de coeficiente fixo de 3ª ordem são apresentadas onde os resíduos $e_3[n]$ são codificados,
15 transmitidos no fluxo e desempacotados no lado de decodificação:

$$e_3[n] = e_3[n] + e_3[n-1]$$

$$e_2[n] = e_2[n] + e_2[n-1]$$

$$e_1[n] = e_1[n] + e_1[n-1]$$

$$s[n] = e_0[n]$$

A predição linear inversa, adaptável ou fixa, executada na etapa 126 é ilustrada para um caso onde o segmento $m+1$ é um segmento de

RAP 900 na Figura 15a e onde o segmento $m+1$ é um segmento transiente 902 na Figura 15b. Um preditor de 5 derivações 904 é usado para reconstruir as amostras de áudio sem perda. Em geral, o preditor recombina 5 amostras reconstruídas sem perda prévias para gerar um valor predito 906 que é adicionado ao resíduo atual 908 para reconstruir sem perda a amostra atual 910. No exemplo de RAP, as primeiras 5 amostras no fluxo de bits de áudio comprimido 912 são amostras de áudio descomprimidas. Conseqüentemente, o preditor pode iniciar decodificação sem perda a segmento $m+1$ sem qualquer história da amostra prévia. Em outras palavras, segmento $m+1$ é um RAP do fluxo de bits. Note, se um transiente também fosse detectado em m segmento $+1$, os parâmetros de predição para m segmento $+1$ e o resto do quadro difeririam daqueles usados em segmentos 1 a m. No exemplo de transiente, todas as amostras em segmentos m e $m+1$ são resíduos, nenhum RAP. Decodificação foi iniciada e a história de predição para o preditor está disponível. Como mostrado, para reconstruir sem perda amostras de áudio em segmentos m e $m+1$, conjuntos diferentes de parâmetros de predição são usados. Para gerar a 1ª amostra sem perda 1 em segmento $m+1$, o preditor usa os parâmetros para segmento $m+1$ usando as últimas cinco amostras reconstruídas sem perda de segmento m. Note, se segmento $m+1$ também fosse um segmento de RAP, as primeiras cinco amostras de segmento $m+1$ seriam amostras originais, não resíduos. Em geral, uma dado quadro pode conter nem um RAP ou transiente, na realidade esse é o resultado mais típico. Alternadamente, um quadro pode incluir um segmento de RAP ou um segmento de transiente ou até mesmo ambos. Um segmento pode ser ambos um RAP e segmento de transiente.

Porque as condições começo de segmento e duração de segmento max são fixadas baseado no local permissível de um RAP desejado ou transiente detectado dentro de um segmento, a seleção da duração ótima de segmento pode gerar um fluxo de bits no qual o RAP desejado ou transiente

detectado de fato se acham dentro de segmentos subsequentes aos segmentos de RAP ou transiente. Isto poderia acontecer se os saltos M e L forem relativamente grandes e a duração ótima de segmento for menos que M e L. O RAP desejado de fato pode se achar em um segmento precedendo o segmento de RAP, mas ainda pode estar dentro da tolerância especificada. As condições sobre tolerância de alinhamento no lado de codificação ainda são mantidas e o decodificador não sabe a diferença. O decodificador acessa simplesmente os segmentos de RAP e transiente.

SEGMENTAÇÃO E SELEÇÃO DE CÓDIGO DE ENTROPIA

O problema de otimização constrangida tratado pelo algoritmo de segmentação adaptável é ilustrado na Figura 16. O problema é codificar um ou mais conjuntos de canal de áudio de multicanal em um fluxo de bits de VBR de tal maneira a minimizar a carga útil de quadro codificado sujeito aos constrangimentos que cada segmento de áudio é decodificável completamente e sem perda com carga útil de segmento codificado menos que um número máximo de bytes. O número máximo de bytes é menos que o tamanho de quadro e tipicamente fixado pelo tamanho máximo de unidade de acesso para ler o fluxo de bits. O problema é ademais constrangido para acomodar acesso aleatório e transientes requerendo que os segmentos sejam selecionados de forma que um RAP desejado deva se achar mais ou menos M blocos de análise do começo do segmento de RAP e um transiente deve se achar dentro dos primeiros L blocos de análise de um segmento. A duração máxima de segmento pode ser ademais constrangida pelo tamanho da memória temporária de saída de decodificador. Neste exemplo, os segmentos dentro de um quadro são constrangidos para serem do mesmo comprimento e uma potência de dois da duração de bloco de análise.

Como mostrado na Figura 16, a duração ótima de segmento para minimizar carga útil de quadro codificado 930 equilibra melhorias em

ganho de predição para um número maior de segmentos de duração mais curta contra o custo de bits adicionais. Neste exemplo, 4 segmentos por quadro provê uma carga útil de quadro menor do que tanto 2 ou 8 segmentos. A solução de dois segmentos é desqualificada porque a carga útil de segmento para o segundo segmento excede o constrangimento de carga útil máxima de segmento 932. A duração de segmento para ambas duas e quatro partições de segmento excede uma duração máxima de segmento 934, que é fixada por alguma combinação, por exemplo, do tamanho da memória temporária de saída de decodificador, local de um ponto de começo de segmento de RAP e/ou local de um ponto começo de segmento de transiente. Conseqüentemente, o algoritmo de segmentação adaptável seleciona os 8 segmentos 936 de duração igual e os parâmetros de predição e codificação de entropia otimizados para essa partição.

Uma concretização exemplar de segmentação e seleção de código de entropia 24 para o caso constrangido (segmentos uniformes, potência de dois de duração de bloco de análise) é ilustrada nas Figuras 7a-b e 8a-b. Para estabelecer a duração ótima de segmento, parâmetros de codificação (seleção e parâmetros de código de entropia) e pares de canais, os parâmetros de codificação e pares de canais são determinados para uma pluralidade de durações de segmento diferentes até a duração máxima de segmento e dentre esses candidatos, o com a mínima carga útil codificada por quadro que satisfaz os constrangimentos que cada segmento deve ser decodificável completamente e sem perda e não exceder um tamanho máximo (número de bytes) é selecionado. A segmentação "ótima", parâmetros de codificação e pares de canais estão certamente sujeitos aos constrangimentos do processo de codificação como também o constrangimento em tamanho de segmento. Por exemplo, no processo exemplar, a duração de tempo de todos os segmentos no quadro é igual, a procura para a duração ótima é executada em uma grade binária começando com uma duração de segmento igual para a

duração de bloco de análise e aumentando por potências de dois, e a seleção de par de canais é válida através do quadro inteiro. Às custas de complexidade de codificador adicional e bits, a duração de tempo pode ser permitida variar dentro de um quadro, a procura para a duração ótima poderia ser solucionada mais finamente e a seleção de par de canais poderia ser feita em uma base por segmento. Neste caso 'constrangido', o constrangimento que assegura que qualquer RAP desejado ou transiente detectado esteja alinhado ao começo de um segmento dentro de uma resolução especificada é concretizado na duração máxima de segmento.

10 O processo exemplar começa iniciando parâmetros de segmento (etapa 150) tal como o número mínimo de amostras em um segmento, o tamanho máximo permitido de carga útil codificada de um segmento, número máximo de segmentos e o máximo número de partições e a duração máxima de segmento. Depois disso, o processamento começa uma malha de partição que é indexada de 0 ao número máximo de partições menos um (etapa 152) e inicia os parâmetros de partição incluindo o número de segmentos, número de amostras em um segmento e o número de bytes consumidos em uma partição (etapa 154). Nesta concretização particular, os segmentos são de duração de tempo igual e o número de escalas de segmentos gradua como uma potência de dois com cada iteração de partição. O número de segmentos é preferivelmente iniciado ao máximo, conseqüentemente duração de tempo mínima, que é igual a um bloco de análise. Porém, o processo poderia usar segmentos de duração de tempo variada, que poderia prover melhor compressão de dados de áudio, mas às custas de custo indireto adicional e complexidade adicional para satisfazer as condições de RAP e transiente. Além disso, o número de segmentos não tem que ser limitado a potências de dois ou procurado da duração mínima à máxima. Neste caso, os pontos de começo de segmento determinados pelo RAP desejado e transientes detectados são constrangimentos adicionais no algoritmo de segmentação

15

20

25

adaptável.

Uma vez iniciado, o processo começa uma malha de conjunto de canal (etapa 156) e determina os parâmetros ótimos de codificação de entropia e seleção de par de canais para cada segmento e o consumo de byte correspondente (etapa 158). Os parâmetros de codificação PWChDecorrFlag [] [], AllChSameParamFlag [] [], RiceCodeFlag [] [] [], CodeParam [] [] [] e ChSetByteCons [] [] são armazenados (etapa 160). Isto é repetido para cada conjunto de canal até que a malha de conjunto de canal termine (etapa 162).

O processo começa uma malha de segmento (etapa 164) e calcula o consumo de byte (SegmByteCons) em cada segmento através de todos os conjuntos de canal (etapa 166) e atualiza o consumo de byte (ByteConsInPart) (etapa 168). Neste momento, o tamanho do segmento (carga útil de segmento codificada em bytes) é comparado ao constrangimento de tamanho máximo (etapa 170). Se o constrangimento for violado, a partição atual é descartada. Além disso, porque o processo começa com a duração de tempo menor, uma vez que um tamanho de segmento seja grande demais, a malha de partição termina (etapa 172) e a melhor solução (duração de tempo, pares de canais, parâmetros de codificação) para esse ponto é empacotada no cabeçalho (etapa 174) e o processo se move para o próximo quadro. Se o constrangimento falhar no tamanho de segmento mínimo (etapa 176), então o processo termina e informa um erro (etapa 178) porque o constrangimento de tamanho máximo não pode ser satisfeito. Assumindo que o constrangimento está satisfeito, este processo é repetido para cada segmento na partição atual até que a malha de segmento termine (etapa 180).

Uma vez que a malha de segmento foi completada e o consumo de byte para o quadro inteiro calculado como representado por ByteConsinPart, esta carga útil é comparada à carga útil mínima atual (MinByteInPart) de uma iteração de partição prévia (etapa 182). Se a partição

atual representar uma melhoria, então a partição atual (PartInd) é armazenada como a partição ótima (OptPartInd) e a carga útil mínima é atualizada (etapa 184). Estes parâmetros e os parâmetros de codificação armazenados são então armazenados como a solução ótima atual (etapa 186). Isto é repetido até que a malha de partição termine com a duração máxima de segmento (etapa 172), a qual momento a informação de segmentação e os parâmetros de codificação são empacotados no cabeçalho (etapa 150) como mostrado nas Figuras 3 e 11a e 11b.

Uma concretização exemplar para determinar os parâmetros ótimos de codificação e consumo de bit associado para um conjunto de canal para uma partição atual (etapa 158) é ilustrada nas Figuras 8a e 8b. O processo começa uma malha de segmento (etapa 190) e malha de canal (etapa 192) na qual os canais para nosso exemplo atual são:

Ch1: L,

Ch2: R

Ch3: R - ChPairDecorrCoeff [1]*L

Ch4: Ls

Ch5: Rs

Ch6: Rs - ChPairDecorrCoeff [2]*Ls

Ch7: C

Ch8: LFE

Ch9: LFE - ChPairDecorrCoeff [3]*C

O processo determina o tipo de código de entropia, parâmetro de codificação correspondente e consumo de bit correspondente para os canais base e correlatado (etapa 194). Neste exemplo, o processo computa parâmetros ótimos de codificação para um código binário e um código de Rice e então seleciona o com o mais baixo consumo de bit para canal e cada segmento (etapa 196). Em geral, a otimização pode ser executada para um, dois ou mais possíveis códigos de entropia. Para os códigos binários, o

número de bits é calculado do valor absoluto max de todas as amostras no segmento do canal atual. O parâmetro de codificação de Rice é calculado do valor absoluto médio de todas as amostras no segmento do canal atual. Baseado na seleção, o RiceCodeFlag é fixado, o BitCons é fixado e o
5 CodeParam é fixado tanto ao NumBitsBinary ou ao RiceKParam (etapa 198).

Se o canal atual sendo processado for um canal correlatado (etapa 200), então a mesma otimização é repetida para o canal descorrelatado correspondente (etapa 202), o melhor código de entropia é selecionado (etapa 204) e os parâmetros de codificação são fixados (etapa 206). O processo se
10 repete até que a malha de canal termine (etapa 208) e a malha de segmento termina (etapa 210).

Neste momento, os parâmetros ótimos de codificação para cada segmento e para cada canal foram determinados. Estes parâmetros de codificação e cargas úteis poderiam ser retornados para os pares de canais
15 (base, correlatado) de áudio de PCM original. Porém, desempenho de compressão pode ser melhorado selecionando entre os canais (base, correlatado) e (base, descorrelatado) nas trincas.

Para determinar quais pares de canal (base, correlatado) ou (base, não correlatado) para as três trincas, uma malha de par de canal é
20 começada (etapa 211) e a contribuição de cada canal correlatado (Ch2, Ch5 e Ch8) e cada canal descorrelatado (Ch3, Ch6 e Ch9) para o consumo de bit de quadro global é calculado (etapa 212). As contribuições de consumo de quadro para cada canal correlatado é comparado contra as contribuições de consumo de quadro para canais descorrelatados correspondentes, isto é, Ch2 a
25 Ch3, Ch5 a Ch6 e Ch8 a Ch9 (etapa 214). Se a contribuição do canal descorrelatado for maior que o canal correlatado, o PWChDecorrFlag é fixado a falso (etapa 216). Caso contrário, o canal correlatado é substituído com o canal descorrelatado (etapa 218) e PWChDecorrFlag é fixado a verdadeiro e os pares de canais são configurados como (base, descorrelatado)

(etapa 220).

Baseado nestas comparações, o algoritmo selecionará:

1. Tanto Ch2 ou Ch3 como o canal que será casado com canal base correspondente Ch1;

5 2. Tanto Ch5 ou Ch6 como o canal que será casado com canal base correspondente Ch4; e

3. Tanto Ch8 ou Ch9 como o canal que será casado com canal base correspondente Ch7.

10 Estas etapas são repetidas para todos os pares de canal até que a malha termine (etapa 222).

Neste momento, os parâmetros ótimos de codificação para cada segmento e cada canal distinto e os pares ótimos de canais foram determinados. Estes parâmetros de codificação para cada par de canal distinto e cargas úteis poderiam ser retornados à malha de partição. Porém, desempenho de compressão adicional pode estar disponível computando um conjunto de parâmetros de codificação globais para cada segmento por todos os canais. Quando muito, a porção de dados codificados da carga útil será o mesmo tamanho como os parâmetros de codificação otimizados para cada canal e provavelmente um pouco maior. Porém, a redução em bits de custo indireto pode mais que compensar a eficiência de codificação dos dados.

20 Usando os mesmos pares de canais, o processo começa uma malha de segmento (etapa 230), calcula os consumos de bits (ChSetByteCons [seg]) por segmento para todos os canais usando os conjuntos distintos de parâmetros de codificação (etapa 232) e armazena ChSetByteCons [seg] (etapa 234). Um conjunto global de parâmetros de codificação (seleção e parâmetros de código de entropia) é então determinado para o segmento por todos os canais (etapa 236) usando o mesmo código binário e cálculos de código de Rice como antes, exceto por todos os canais. Os melhores parâmetros são selecionados e o consumo de byte (SegmByteCons) é

calculado (etapa 238). O SegmByteCons é comparado ao CHSetByteCons [seg] (etapa 240). Se usar parâmetros globais não reduz consumo de bit, o AllChSamParamFlag[seg] é fixado a falso (etapa 242). Caso contrário, o AllChSameParamFlag [seg] é fixado a verdadeiro (etapa 244) e os parâmetros de codificação globais e consumo de bit correspondente por segmento são salvos (etapa 246). Este processo se repete até que o fim da malha de segmento seja alcançado (etapa 248). O processo inteiro se repete até que a malha de conjunto de canal termine (etapa 250).

O processo de codificação é estruturado de um modo que funcionalidade diferente pode ser desabilitada pelo controle de alguns indicadores. Por exemplo, um único indicador controla se a análise de decorrelação de canal em pares será executada ou não. Outro indicador controla se a análise de predição adaptável (ainda outro indicador para predição fixa) é para ser executada ou não. Além disso, um único indicador controla se a procura por parâmetros globais através de todos os canais é para ser executada ou não. Segmentação também é controlável fixando o número de partições e duração mínima de segmento (na forma mais simples pode ser uma única partição com duração de segmento predeterminada). Um indicador indica a existência de um segmento de RAP e outro indicador indica a existência de um segmento transiente. Em essência, fixando alguns indicadores no codificador, o codificador pode desmornar a moldando simples e codificação de entropia.

CODEC DE ÁUDIO SEM PERDA COMPATÍVEL REVERSO

O codec sem perda pode ser usado como um "codificador de extensão" em combinação com um codificador de núcleo com perda. Um fluxo de código de núcleo "com perda" é empacotado como um fluxo de bits de núcleo e um sinal de diferença codificado sem perda é empacotado como um fluxo de bits de extensão separado. Ao decodificar em um decodificador

com características sem perda estendidas, os fluxos com perda e sem perda são combinados para construir um sinal reconstruído sem perda. Em um decodificador de geração anterior, o fluxo sem perda é ignorado, e o fluxo "com perda" de núcleo é decodificado para prover um de sinal de áudio de multicanal de alta qualidade com as características de largura de banda e relação de sinal para ruído do fluxo de núcleo.

Figura 9 mostra uma vista em nível de sistema de um codificador sem perda compatível reverso 400 para um canal de um sinal de multicanal. Um sinal de áudio digitalizado, apropriadamente amostras de áudio de PCM de M bits, é provido na entrada 402. Preferivelmente, o sinal de áudio digitalizado tem uma taxa de amostragem e largura da banda que excede aquela de um codificador de núcleo com perda 404 modificado. Em uma concretização, a taxa de amostragem do sinal de áudio digitalizado é 96 kHz (correspondendo a uma largura de banda de 48 kHz para o áudio amostrado). Também deveria ser entendido que o áudio de entrada pode ser, e preferivelmente é, um sinal de multicanal em que cada canal é amostrado a 96 kHz. A discussão que segue se concentrará no processamento de um único canal, mas a extensão para múltiplos canais é direta. O sinal de entrada é duplicado a nó 406 e operado em ramais paralelos. Em um primeiro ramal do caminho de sinal, um codificador de banda larga com perda modificado 404 codifica o sinal. O codificador de núcleo modificado 404, que é descrito em detalhes abaixo, produz um fluxo de bits de núcleo codificado 408, que é levado a um empacotador ou multiplexador 410. O fluxo de bits de núcleo 408 também é comunicado a um decodificador de núcleo modificado 412, que produz como saída um sinal de núcleo reconstruído modificado 414.

Enquanto isso, o sinal de áudio digitalizado de entrada 402 no caminho paralelo sofre um atraso de compensação 416, substancialmente igual ao atraso introduzido no fluxo de áudio reconstruído (por codificação modificada e decodificadores modificados), para produzir um fluxo de áudio

digitalizado atrasado. O fluxo de áudio 400 é subtraído do fluxo de áudio digitalizado atrasado 414 a nó de adição 420.

Nó de adição 420 produz um sinal de diferença 422 que representa o sinal original e o sinal de núcleo reconstruído. Para realizar codificação puramente "sem perda", é necessário codificar e transmitir o sinal de diferença com técnicas de codificação sem perda. Por conseguinte, o sinal de diferença 422 é codificado com um codificador sem perda 424, e o fluxo de bits de extensão 426 é empacotado com o fluxo de bits de núcleo 408 em empacotador 410 para produzir um fluxo de bits de saída 428.

Note que a codificação sem perda produz um fluxo de bits de extensão 426 que está a uma taxa de bit variável, para acomodar as necessidades do codificador sem perda. O fluxo empacotado é então opcionalmente sujeito a camadas adicionais de codificação incluindo codificação de canal, e então transmitido ou gravado. Note que para propósitos desta exposição, gravação pode ser considerado como transmissão por um canal.

O codificador de núcleo 404 é descrito como "modificado" porque em uma concretização capaz de operar largura de banda estendida, o codificador de núcleo requereria modificação. Um banco de filtro de análise de 64 bandas 430 dentro do codificador descarta a metade de seus dados de saída 432 e um codificador de sub-banda de núcleo 434 codifica só as bandas de frequência mais baixas 32. Esta informação descartada não é de nenhuma preocupação para decodificadores de legado que seriam incapazes de reconstruir a metade superior do espectro de sinal em qualquer caso. A informação restante é codificada como pelo codificador inalterado para formar um fluxo de saída de núcleo compatível reverso. Porém, em outra concretização operando a ou abaixo de taxa de amostragem de 48 kHz, o codificador de núcleo poderia ser uma versão substancialmente inalterada de um codificador de núcleo anterior. Semelhantemente, para operação acima da

taxa de amostragem de decodificadores de legado, o decodificador de núcleo modificado 412 inclui um decodificador de sub-banda de núcleo 436 que decodifica amostras nas 32 sub-bandas inferiores. O decodificador de núcleo modificado leva as amostras de sub-banda das 32 sub-bandas inferiores e zeros fora das amostras de sub-banda não transmitidas para as 32 bandas superiores 438 e reconstrói todas as 64 bandas usando um filtro de síntese de QMF de 64 bandas 440. Para operação à taxa de amostragem convencional (por exemplo, 48 kHz e abaixo), o decodificador de núcleo poderia ser uma versão substancialmente inalterada de um decodificador de núcleo anterior ou equivalente. Em algumas concretizações, a escolha de taxa de amostragem poderia ser feita na hora de codificação, e os módulos de codificação e decodificação reconfigurados nesse momento por software como desejado.

Desde que o codificador sem perda está sendo usado para codificar o sinal de diferença, pode parecer que um código de entropia simples bastaria. Porém, por causa das limitações de taxa de bit nos codecs de núcleo com perda existentes, uma quantidade considerável dos bits totais exigidos para prover um fluxo de bits sem perda ainda permanece. Além disso, por causa das limitações de largura de banda do codec de núcleo, o conteúdo de informação acima de 24 kHz no sinal de diferença ainda é correlatado. Por exemplo, muitos dos componentes harmônicos incluindo trompete, violão, triângulo, ..., alcançam bem além de 30 kHz). Portanto, codecs sem perda mais sofisticados que melhoram desempenho compressão adicionam valor. Além disso, em algumas aplicações, os fluxos de bits de núcleo e extensão ainda devem satisfazer o constrangimento que as unidades decodificáveis não devem exceder um tamanho máximo. O codec sem perda da presente invenção provê ambos desempenho melhorado de compressão e flexibilidade melhorada para satisfazer estes constrangimentos.

Por meio de exemplo, 8 canais de áudio de PCM de 96 kHz de 24 bits requer 18,5 Mbps. Compressão sem perda pode reduzir isto a cerca de

9 Mbps. Acústica Coerente de DTS codificaria o núcleo a 1,5 Mbps, deixando um sinal de diferença de 7,5 Mbps. Para tamanho de segmento de 2 kByte max, a duração média de segmento é $2048 \times 8 / 7500000 = 2,18$ ms ou aproximadamente 209 amostras a 96 kHz. Um tamanho de quadro típico para o núcleo com perda para satisfazer o tamanho max está entre 10 e 20 ms.

A um nível de sistema, o codec sem perda e o codec sem perda compatível reverso podem ser combinados para codificar sem perda canais de áudio extras a uma largura de banda estendida enquanto mantendo compatibilidade reversa com codecs com perda existentes. Por exemplo, 8 canais de áudio de 96 kHz a 18,5 Mbps pode ser codificado sem perda para incluir 5.1 canais de áudio de 48 kHz a 1,5 Mbps. O codificador de núcleo mais sem perda seria usado para codificar os 5.1 canais. O codificador sem perda será usado para codificar os sinais de diferença nos 5.1 canais. Os 2 canais restantes são codificados em um conjunto de canal separado usando o codificador sem perda. Desde que todos os conjuntos de canal precisam ser considerados ao tentar otimizar duração de segmento, todas as ferramentas de codificação serão usados de uma maneira ou de outra. Um decodificador compatível decodificaria todos os 8 canais e reconstruiria sem perda o sinal de áudio de 96 kHz a 18,5 Mbps. Um decodificador mais antigo decodificaria só os 5.1 canais e reconstruiria o 48 kHz a 1,5Mbps.

Em geral, mais de um conjunto de canal sem perda puro pode ser provido para o propósito de graduar a complexidade do decodificador. Por exemplo, para uma mistura original 10.2, os conjuntos de canal poderiam ser arrançados tal que:

- 25 - CHSET1 leva 5.1 (com mistura abaixo de 10.2 a 5.1 embutido) e é codificado usando núcleo+sem perda;
- CHSET1 e CHSET2 levam 7.1 (com mistura abaixo de 10.2 a 7.1 embutido), onde CHSET2 codifica 2 canais usando sem perda;
- CHSET1+CHSET2+CHSET3 levam mistura 10.2 discreta

completa, onde CHSET3 codifica 3.1 canais restantes usando só sem perda.

Um decodificador que é capaz de decodificar apenas 5.1 decodificará só CHSET1 e ignorará todos os outros conjuntos de canal. Um decodificador que é capaz de decodificar apenas 7.1 decodificará CHSET1 e CHSET2 e ignorará todos os outros conjuntos de canal.

Além disso, o núcleo com perda mais sem perda não está limitado a 5.1. Implementações atuais estão até 6.1 usando com perda (núcleo+XCh) e sem perda e pode suportar m.n canais genéricos organizados em qualquer número de conjuntos de canal. A codificação com perda terá um núcleo compatível reverso a 5.1 e todos os outros canais que são codificados com o codec com perda entrarão na extensão de XXCh. Isto provê o codec sem perda global com flexibilidade de projeto considerável para permanecer compatível reverso com decodificadores existentes enquanto suportam canais adicionais.

Enquanto várias concretizações ilustrativas da invenção foram mostradas e descritas, numerosas variações e concretizações alternadas ocorrerão àqueles qualificados na arte. Tais variações e concretizações alternadas são contempladas, e podem ser feitas sem partir do espírito e extensão da invenção como definida nas reivindicações anexas.

transiente; e

para cada canal no conjunto de canal, aplicar tanto o único conjunto de coeficientes de predição aos sinais de áudio comprimidos para todos os segmentos no quadro para reconstruir sem perda áudio de PCM ou aplicar o primeiro conjunto de coeficientes de predição aos sinais de áudio comprimidos começando no primeiro segmento e aplicar o segundo conjunto de coeficientes de predição aos sinais de áudio comprimidos começando no segmento de transiente.

23. Método de acordo com a reivindicação 16, caracterizado pelo fato de que o fluxo de bits ademais inclui informação de cabeçalho de conjunto de canal incluindo um indicador de descorrelação de canal em pares, uma ordem de canal original, e coeficientes de descorrelação de canal quantizados, dita reconstrução gerando áudio de PCM descorrelatado, o método ademais incluindo:

desempacotar o cabeçalho para extrair a ordem de canal original, o indicador de descorrelação de canal em pares e os coeficientes de descorrelação de canal quantizados e executar uma descorrelação de canal cruzada inversa para reconstruir áudio de PCM para cada canal de áudio em dito conjunto de canal.

24. Método de acordo com a reivindicação 23, caracterizado pelo fato de que o indicador de descorrelação de canal em pares indica se um primeiro par de canais incluindo um canal base e um correlatado ou um segundo par de canais incluindo o canal base e um descorrelatado para uma trinca incluindo os canais base, correlatado e descorrelatado era codificado, o método ademais incluindo:

se o indicador indicar um segundo par de canais, multiplicar o canal base pelo coeficiente de descorrelação de canal quantizado e adicioná-lo ao canal descorrelatado para gerar áudio de PCM no canal correlatado.

25. Um ou mais meios legíveis por computador caracterizado

pelo fato de incluir instruções executáveis por computador que, quando executadas, executam o método como recitado de acordo com a reivindicação 16.

5 26. Um ou mais dispositivos semicondutores caracterizado pelo fato de incluir circuitos digitais configurados para executar o método como recitado de acordo com a reivindicação 16.

27. Método para codificar áudio de multicanal em um fluxo de bits de áudio de taxa de bit variável (VBR) sem perda, caracterizado pelo fato de incluir:

10 formar em blocos o áudio de multicanal incluindo pelo menos um conjunto de canal em quadros de duração de tempo igual, cada quadro incluindo um cabeçalho e uma pluralidade de segmentos, que cada dito segmento tendo uma duração de um ou mais blocos de análise;

para cada quadro sucessivo,

15 detectar a existência de um transiente em um bloco de análise de transiente no quadro para cada canal do conjunto de canal;

dividir o quadro de forma que qualquer bloco de análise de transiente esteja localizado dentro dos primeiros L blocos de análise de um segmento nos seus canais correspondentes;

20 determinar um primeiro conjunto de parâmetros de predição para segmentos antes e não incluindo o bloco de análise de transiente e um segundo conjunto de parâmetros de predição para segmentos incluindo e subsequente ao bloco de análise de transiente para cada canal no conjunto de canal;

25 comprimir os dados de áudio usando o primeiro e segundo conjuntos de parâmetros de predição na primeira e segunda partições, respectivamente, para gerar sinais de áudio residuais;

determinar uma duração de segmento e parâmetros de codificação de entropia para cada segmento das amostras de áudio residuais

para reduzir uma carga útil codificada dimensionada variável do quadro sujeito a constrangimentos que cada segmento deve ser decodificável completamente e sem perda, ter uma duração menos que a duração de quadro e ter uma carga útil de segmento codificada menos que um número máximo de bytes menos que o tamanho de quadro;

empacotar informação de cabeçalho incluindo duração de segmento, parâmetros de transiente indicando a existência e local do transiente, parâmetros de predição, parâmetros de codificação de entropia e dados de navegação de fluxo de bits no cabeçalho de quadro no fluxo de bits;

10 e

empacotar os dados de áudio comprimidos e codificados em entropia para cada segmento nos segmentos de quadro no fluxo de bits.

28. Método de acordo com a reivindicação 27, caracterizado pelo fato de incluir para cada canal no conjunto de canal:

15 determinar um terceiro conjunto de parâmetros de predição para o quadro inteiro;

comprimir os dados de áudio usando o terceiro conjunto de parâmetros de predição no quadro inteiro para gerar sinais de áudio residuais;

e

20 selecionar tanto o terceiro conjunto ou primeiro e segundo conjuntos de parâmetros de predição baseado em uma medida de eficiência de codificação dos seus sinais de áudio residuais respectivos,

em que se dito terceiro conjunto for selecionado, desabilitar o constrangimento em duração de segmento relativo a local do transiente dentro de L blocos de análise do começo de um segmento.

25 29. Método de acordo com a reivindicação 27, caracterizado pelo fato de incluir:

receber um código de temporização que especifica pontos de acesso aleatório (RAPs) desejado no fluxo de bits de áudio;

determinar até um bloco de análise de RAP dentro do quadro do código de temporização;

fixar o começo de um segmento de RAP de forma que o bloco de análise de RAP se ache dentro de M blocos de análise do começo;

5 considerar o limite de segmento imposto pelo segmento de RAP ao dividir o quadro para determinar o primeiro e segundo conjuntos de parâmetros de predição;

desabilitar dita predição para as primeiras amostras até a ordem de predição seguindo o começo do segmento de RAP para gerar amostras de áudio originais precedidas e/ou seguidas por amostras de áudio residuais para dito primeiro e segundo, e terceiro conjuntos de parâmetros de predição;

determinar a duração de segmento que reduz carga útil de quadro codificado enquanto satisfazendo os constrangimentos que um bloco de análise de RAP se acha dentro de M blocos de análise do começo do segmento de RAP e/ou blocos de análise de transiente devem se achar dentro dos primeiros L blocos de análise de um segmento; e

empacotar parâmetros de RAP indicando a existência e local do RAP e dados de navegação de fluxo de bits no cabeçalho de quadro.

20 30. Método de acordo com a reivindicação 27, caracterizado pelo fato de incluir:

usar o local detectado do bloco de análise de transiente para determinar uma duração máxima de segmento como uma potência de dois da duração de bloco de análise tal que dito transiente se ache dentro dos primeiros L blocos de análise de um segmento,

25 em que uma duração de segmento uniforme que é uma potência de dois da duração de bloco de análise e não excede a duração máxima de segmento é determinada para reduzir carga útil de quadro codificado sujeito aos constrangimentos.

31. Método de acordo com a reivindicação 30, caracterizado pelo fato de que a duração máxima de segmento é ademais constrangida pelo tamanho da memória temporária de saída disponível em um decodificador.

5 32. Método de acordo com a reivindicação 27, caracterizado pelo fato de que o número máximo de bytes para a carga útil de segmento codificado é imposto por um constrangimento de tamanho de unidade de acesso do fluxo de bits de áudio.

10 33. Método de acordo com a reivindicação 27, caracterizado pelo fato de que dito fluxo de bits inclui primeiro e segundo conjuntos de canal, dito método selecionando primeiro e segundo conjuntos de parâmetros de predição para cada canal em cada conjunto de canal baseado na detecção de transientes em locais diferentes para pelo menos um canal nos conjuntos de canal respectivos, em que dita duração de segmento é determinada de forma que cada dito transiente se ache dentro dos primeiros L blocos de análise de
15 um segmento no qual o transiente ocorre.

34. Método de acordo com a reivindicação 33, caracterizado pelo fato de que o primeiro conjunto de canal inclui áudio de multicanal 5.1 e o segundo conjunto de canal inclui pelo menos um canal de áudio adicional.

20 35. Método de acordo com a reivindicação 27, caracterizado pelo fato de que os parâmetros de transiente incluem um indicador de transiente indicando a existência de um transiente e um ID de transiente indicando o número de segmento no qual o transiente ocorre.

25 36. Método de acordo com a reivindicação 27, caracterizado pelo fato de incluir gerar um canal descorrelatado para pares de canais formarem uma trinca incluindo um canal base, correlatado e descorrelatado, selecionar tanto um primeiro par de canais incluindo um canal base e um correlatado ou um segundo par de canais incluindo um canal base e um descorrelatado, e codificar em entropia os canais nos pares de canais selecionados.

37. Método de acordo com a reivindicação 36, caracterizado pelo fato de que os pares de canais são selecionados:

se a variância do canal descorrelatado for menor que a variância do canal correlatado por um limiar, selecionar o segundo par de
5 canais antes de determinar duração de segmento; e

caso contrário, adiar seleção do primeiro ou segundo par de canais até determinação de duração de segmento baseada na qual par de canais contribui os menos bits à carga útil codificada.

38. Um ou mais meios legíveis por computador caracterizado pelo fato de incluir instruções executáveis por computador que, quando executadas, executam o método como recitado de acordo com a reivindicação
10 27.

39. Um ou mais dispositivos semicondutores caracterizado pelo fato de incluir circuitos digitais configurados para executar o método
15 como recitado de acordo com a reivindicação 27.

40. Método para decodificar um fluxo de áudio de multicanal de taxa de bit variável (VBR) sem perda, caracterizado pelo fato de incluir:

receber um fluxo de bits de áudio de multicanal VBR sem perda como uma seqüência de quadros divididos em uma pluralidade de
20 segmentos tendo uma carga útil de quadro de comprimento variável e incluindo pelo menos um conjunto de canal decodificável independentemente e reconstruível sem perda incluindo uma pluralidade de canais de áudio para um sinal de áudio de multicanal, cada quadro incluindo informação de cabeçalho incluindo duração de segmento, informação de cabeçalho de
25 conjunto de canal incluindo parâmetros de transiente que indicam a existência e local de um segmento de transiente em cada canal, coeficientes de predição para cada dito canal incluindo um único conjunto de coeficientes de predição baseados em quadro se nenhum transiente estiver presente e primeiro e segundo conjuntos de coeficientes de predição baseados em partição se um

transiente estiver presente em cada dito conjunto de canal, e informação de cabeçalho de segmento para cada dito conjunto de canal incluindo pelo menos um indicador de código de entropia e pelo menos um parâmetro de codificação de entropia, e sinais de áudio de multicanal comprimidos codificados em entropia armazenados em dito número de segmentos;

desempacotar o cabeçalho para extrair a duração de segmento;

desempacotar o cabeçalho para o pelo menos um dito conjunto de canal para extrair o indicador de código de entropia e parâmetro de codificação e os sinais de áudio de multicanal comprimidos codificados em entropia para cada segmento e executar uma decodificação de entropia em cada segmento usando o código de entropia selecionado e parâmetro de codificação para gerar sinais de áudio comprimidos para cada segmento;

desempacotar o cabeçalho para o pelo menos um dito conjunto de canal para extrair os parâmetros de transiente para determinar a existência e local de segmentos de transiente em cada canal no conjunto de canal;

desempacotar o cabeçalho para o pelo menos um dito conjunto de canal para extrair o único conjunto de coeficientes de predição baseados em quadro ou primeiro e segundo conjuntos de coeficientes de predição baseados em partição para cada canal dependendo da existência de um transiente; e

para cada canal no conjunto de canal, aplicar tanto o único conjunto de coeficientes de predição aos sinais de áudio comprimidos para todos os segmentos no quadro para reconstruir sem perda áudio de PCM ou aplicar o primeiro conjunto de coeficientes de predição aos sinais de áudio comprimidos começando no primeiro segmento e aplicar o segundo conjunto de coeficientes de predição aos sinais de áudio comprimidos começando no segmento de transiente.

41. Método de acordo com a reivindicação 40, caracterizado pelo fato de que o fluxo de bits ademais inclui informação de cabeçalho de

conjunto de canal incluindo um indicador de descorrelação de canal em pares, uma ordem de canal original, e coeficientes de descorrelação de canal quantizados, dita reconstrução gerando áudio de PCM descorrelatado, o método ademais incluindo:

5 desempacotar o cabeçalho para extrair a ordem de canal original, o indicador de descorrelação de canal em pares e os coeficientes de descorrelação de canal quantizados e executar uma descorrelação de canal cruzada inversa para reconstruir áudio de PCM para cada canal de áudio em dito conjunto de canal.

10 42. Método de acordo com a reivindicação 41, caracterizado pelo fato de que o indicador de descorrelação de canal em pares indica se um primeiro par de canais incluindo um canal base e um correlatado ou um segundo par de canais incluindo um canal base e um descorrelatado para uma trinca incluindo o canal base, correlatado e descorrelatado era codificado, o
15 método ademais incluindo:

se o indicador indicar um segundo par de canais, multiplicar o canal base pelo coeficiente de descorrelação de canal quantizado e adicioná-lo ao canal descorrelatado para gerar áudio de PCM no canal correlatado.

20 43. Método de acordo com a reivindicação 40, caracterizado pelo fato de incluir:

receber um quadro tendo informação de cabeçalho incluindo parâmetros de RAP que indicam a existência e local de até um segmento de RAP e dados de navegação;

25 desempacotar o cabeçalho do próximo quadro no fluxo de bits para extrair os parâmetros de RAP, se tentar iniciar decodificação a RAP, saltar ao próximo quadro até que um quadro tendo um segmento de RAP seja detectado e usar os dados de navegação para navegar ao começo do segmento de RAP; e

quando um segmento de RAP é encontrado, desabilitar a

predição para as primeiras amostras de áudio até a ordem de predição para reconstruir sem perda o áudio de PCM.

5 44. Método de acordo com a reivindicação 40, caracterizado pelo fato de que o número e duração de segmentos varia quadro a quadro para minimizar a carga útil de comprimento variável de cada quadro sujeito a constrangimentos que a carga útil de segmento codificada seja menos que um número máximo de bytes menos que o tamanho de quadro e reconstruível sem perda.

10 45. Um ou mais meios legíveis por computador caracterizado pelo fato de incluir instruções executáveis por computador que, quando executadas, executam o método como recitado de acordo com a reivindicação 40.

15 46. Um ou mais dispositivos semicondutores caracterizado pelo fato de que incluem circuitos digitais configurados para executar o método como recitado de acordo com a reivindicação 40.

47. Decodificador de áudio de multicanal para iniciar decodificação de um fluxo de bits de áudio de multicanal de taxa de bit variável (VBR) sem perda a um ponto de acesso aleatório (RAP), caracterizado pelo fato de que dito decodificador é configurado para:

20 receber um fluxo de bits de áudio multicanal de VBR sem perda como uma seqüência de quadros divididos em uma pluralidade de segmentos tendo uma carga útil de quadro de comprimento variável e incluindo pelo menos um conjunto de canal decodificável independentemente e reconstruível sem perda incluindo uma pluralidade de canais de áudio para
25 um sinal de áudio de multicanal, cada quadro incluindo informação de cabeçalho incluindo duração de segmento, parâmetros de RAP que indicam a existência e local de até um segmento de RAP, dados de navegação, informação de cabeçalho de conjunto de canal incluindo coeficientes de predição para cada dito canal em cada dito conjunto de canal, e informação de

cabeçalho de segmento para cada dito conjunto de canal incluindo pelo menos um indicador de código de entropia e pelo menos um parâmetro de codificação de entropia, e sinais de áudio de multicanal comprimidos codificados em entropia armazenados em dito número de segmentos;

5 desempacotar o cabeçalho do próximo quadro no fluxo de bits para extrair os parâmetros de RAP até que um quadro tendo um segmento de RAP seja detectado;

10 desempacotar o cabeçalho do quadro selecionado para extrair a duração de segmento e dados de navegação para navegar ao começo do segmento de RAP;

15 desempacotar o cabeçalho para o pelo menos um dito conjunto de canal para extrair o indicador de código de entropia e parâmetro de codificação e os sinais de áudio de multicanal comprimidos codificados em entropia e executar uma decodificação de entropia no segmento de RAP usando o código de entropia selecionado e parâmetro de codificação para gerar sinais de áudio comprimidos para o segmento de RAP; e

20 desempacotar o cabeçalho para o pelo menos um dito conjunto de canal para extrair coeficientes de predição e reconstruir os sinais de áudio comprimidos para reconstruir sem perda áudio de PCM para cada canal de áudio em dito conjunto de canal para o segmento de RAP; e

 decodificar o resto dos segmentos no quadro e quadros subseqüentes em ordem.

25 48. Decodificador de áudio de multicanal para decodificar um fluxo de bits de áudio de multicanal de taxa de bit variável (VBR) sem perda, caracterizado pelo fato de que dito decodificador é configurado para:

 receber um fluxo de bits de áudio de multicanal de VBR sem perda como uma seqüência de quadros divididos em uma pluralidade de segmentos tendo uma carga útil de quadro de comprimento variável e incluindo pelo menos um conjunto de canal decodificável independentemente

e reconstruível sem perda incluindo uma pluralidade de canais de áudio para um sinal de áudio de multicanal, cada quadro incluindo informação de cabeçalho incluindo duração de segmento, informação de cabeçalho de conjunto de canal incluindo parâmetros de transiente que indicam a existência e local de um segmento de transiente em cada canal, coeficientes de predição para cada dito canal incluindo um único conjunto de coeficientes de predição baseados em quadro se nenhum transiente estiver presente e primeiro e segundo conjuntos de coeficientes de predição baseados em partição se um transiente estiver presente em cada dito conjunto de canal, e informação de cabeçalho de segmento para cada dito conjunto de canal incluindo pelo menos um indicador de código de entropia e pelo menos um parâmetro de codificação de entropia, e sinais de áudio de multicanal comprimidos codificados em entropia armazenados em dito número de segmentos;

desempacotar o cabeçalho para extrair a duração de segmento;

desempacotar o cabeçalho para o pelo menos um dito conjunto de canal para extrair o indicador de código de entropia e parâmetro de codificação e os sinais de áudio de multicanal comprimidos codificados em entropia para cada segmento e executar uma decodificação de entropia em cada segmento usando o código de entropia selecionado e parâmetro de codificação para gerar sinais de áudio comprimidos para cada segmento;

desempacotar o cabeçalho para o pelo menos um dito conjunto de canal para extrair os parâmetros de transiente para determinar a existência e local de segmentos de transiente em cada canal no conjunto de canal;

desempacotar o cabeçalho para o pelo menos um dito conjunto de canal para extrair o único conjunto de coeficientes de predição baseados em quadro ou primeiro e segundo conjuntos de coeficientes de predição baseado em partição para cada canal dependendo da existência de um transiente; e

para cada canal no conjunto de canal, aplicar tanto o único

conjunto de coeficientes de predição aos sinais de áudio comprimidos para todos os segmentos no quadro para reconstruir sem perda áudio de PCM ou aplicar o primeiro conjunto de coeficientes de predição aos sinais de áudio comprimidos começando no primeiro segmento e aplicar o segundo conjunto de coeficientes de predição aos sinais de áudio comprimidos começando no segmento de transiente.

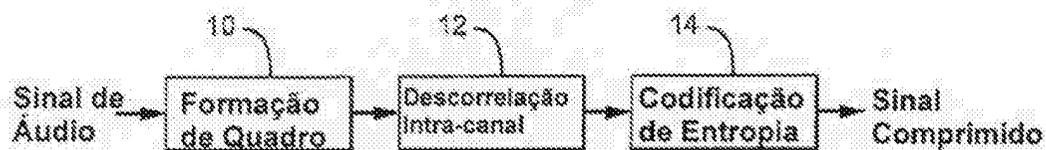


FIG. 1
TÉCNICA ANTERIOR

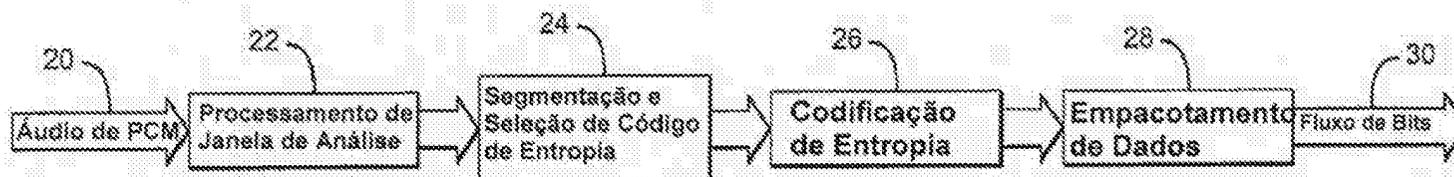
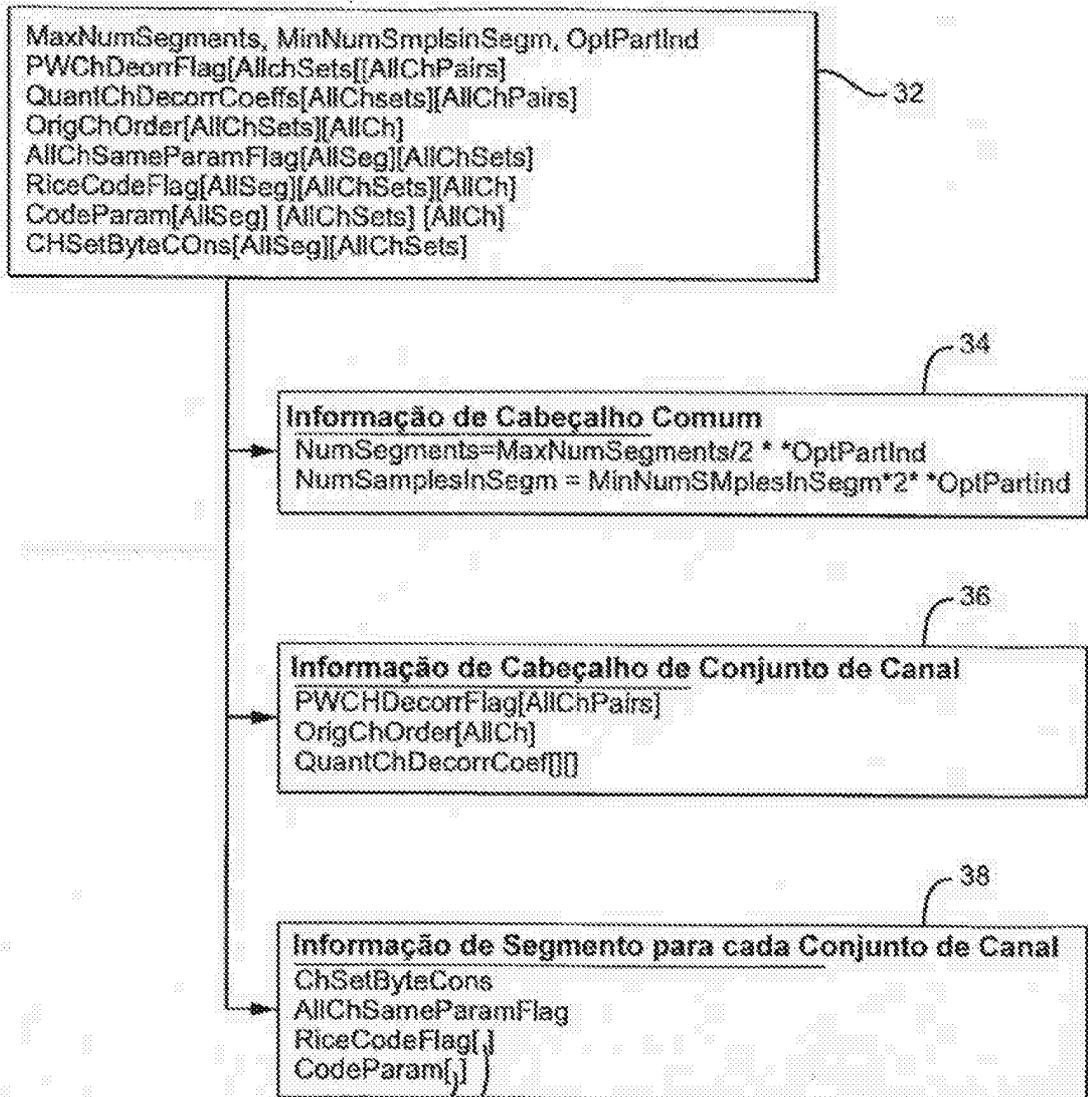


FIG. 2a



FIG. 2b



Se AllChSameParamFlag = verdadeiro

- RiceCodeFlag único é transmitido por conjunto de canal
- CodeParam único é transmitido por conjunto de canal

Senão

- RiceCodeFlag é transmitido para cada canal no conjunto de canal
- RiceCodeFlag é transmitido para cada canal no conjunto de canal

FIG. 3

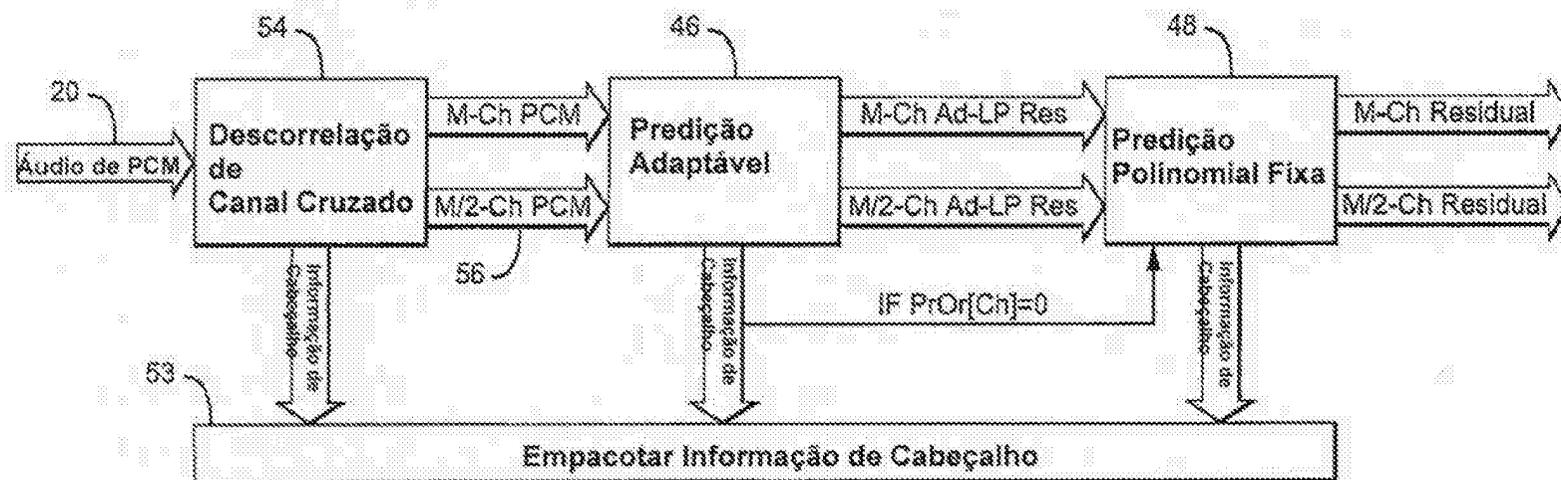


FIG. 4a

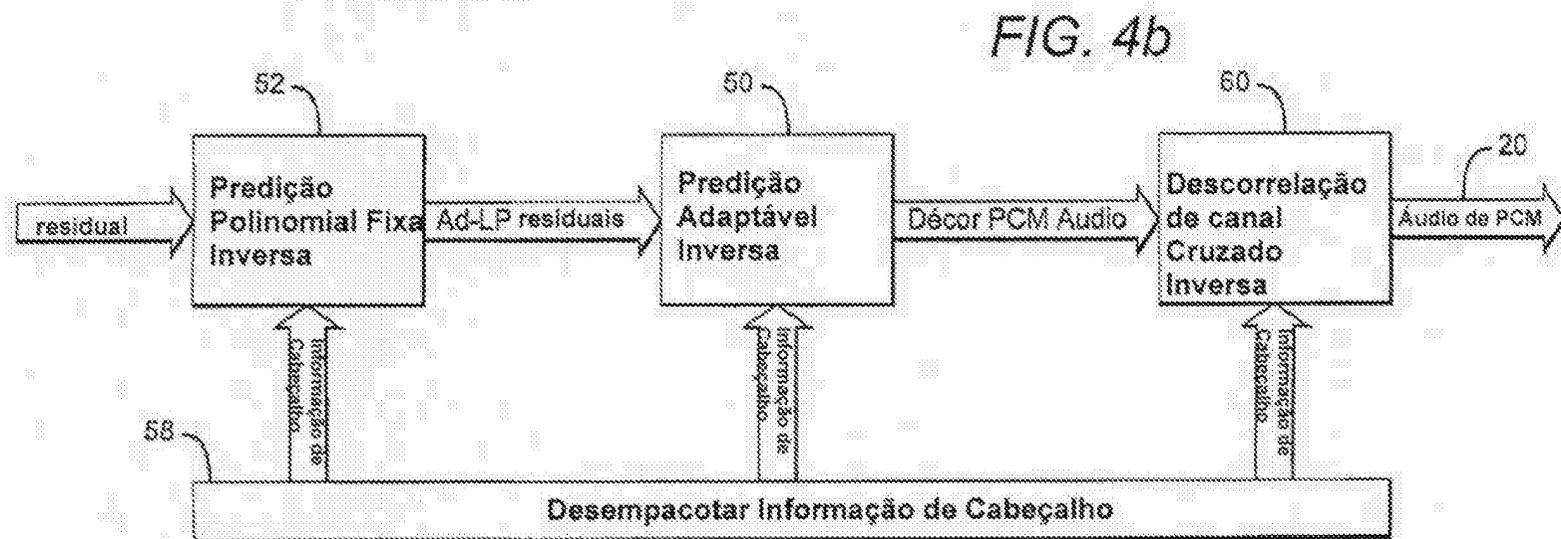


FIG. 4b

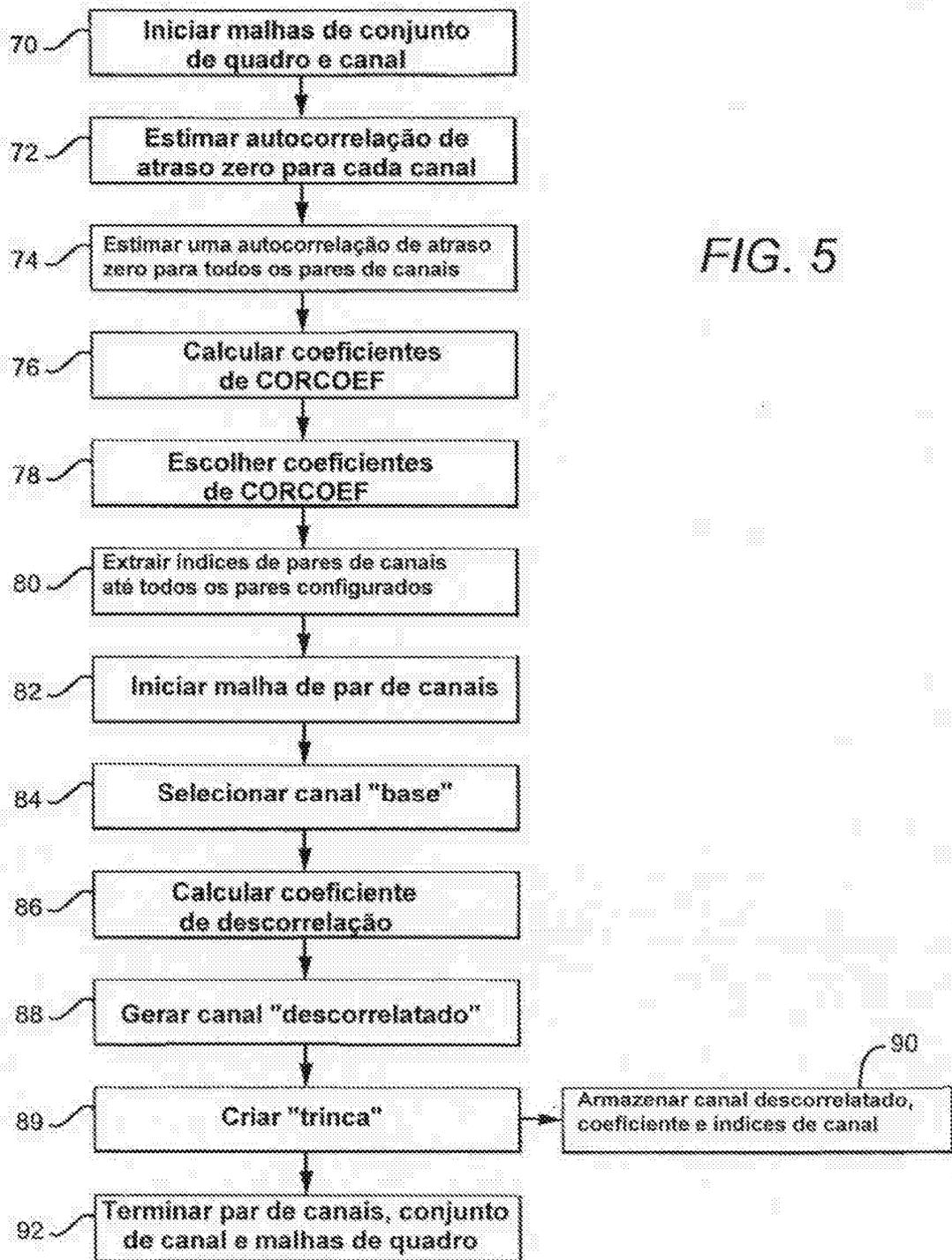


FIG. 5

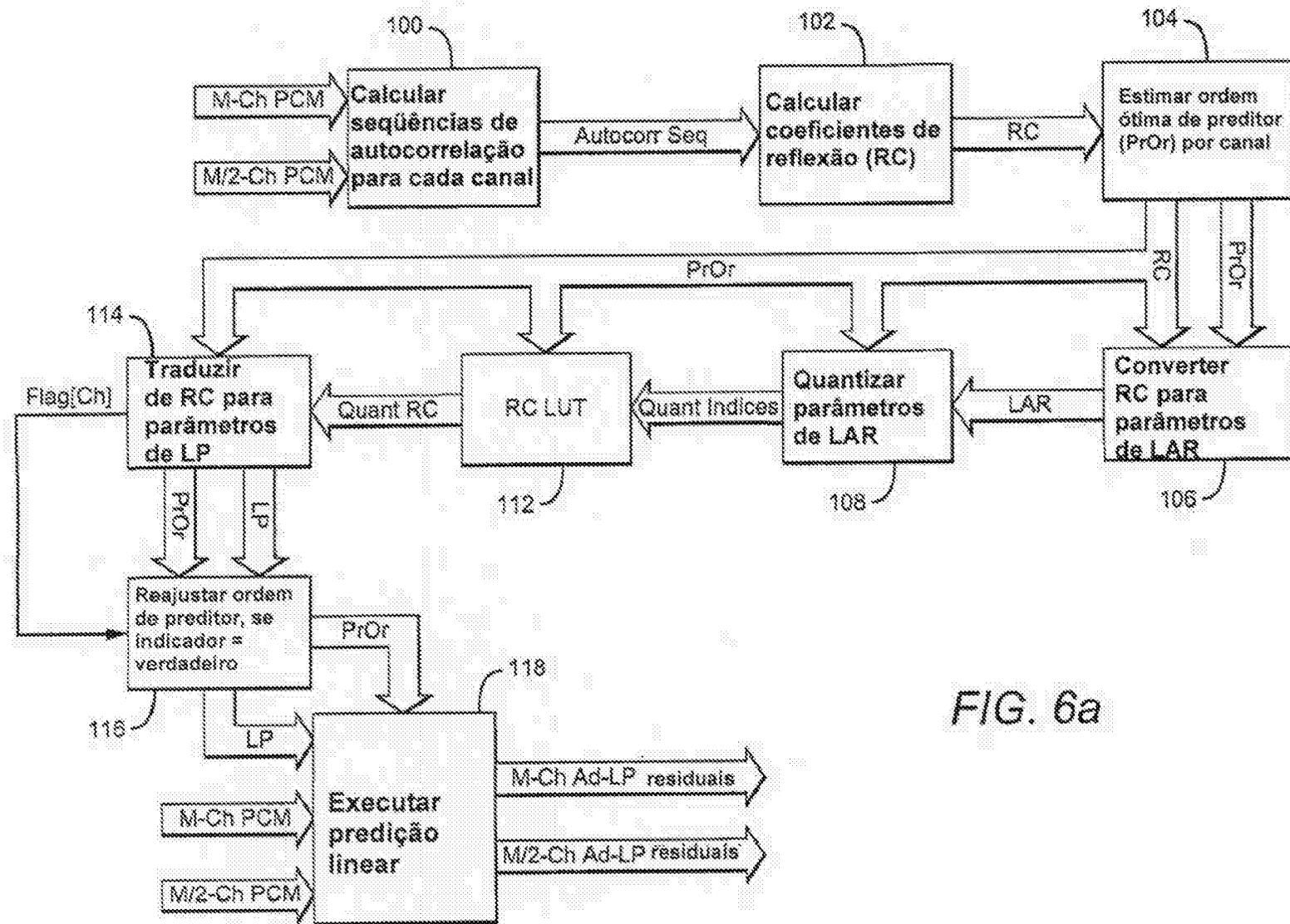
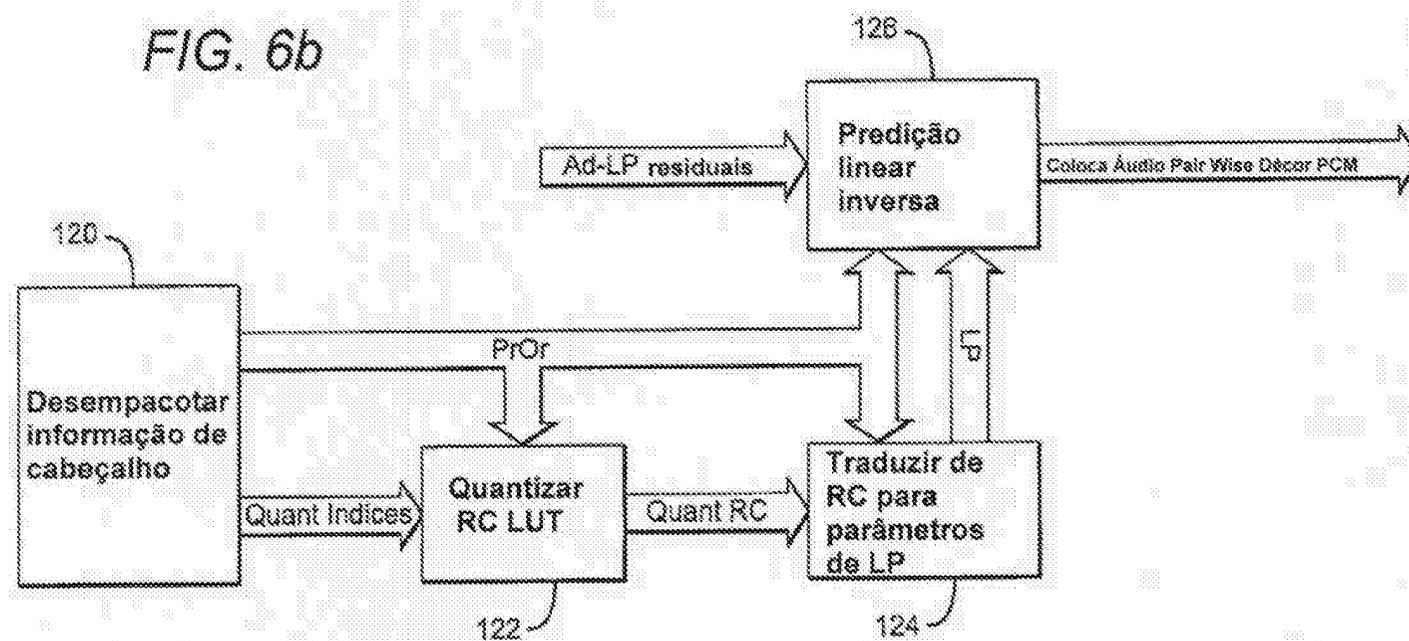
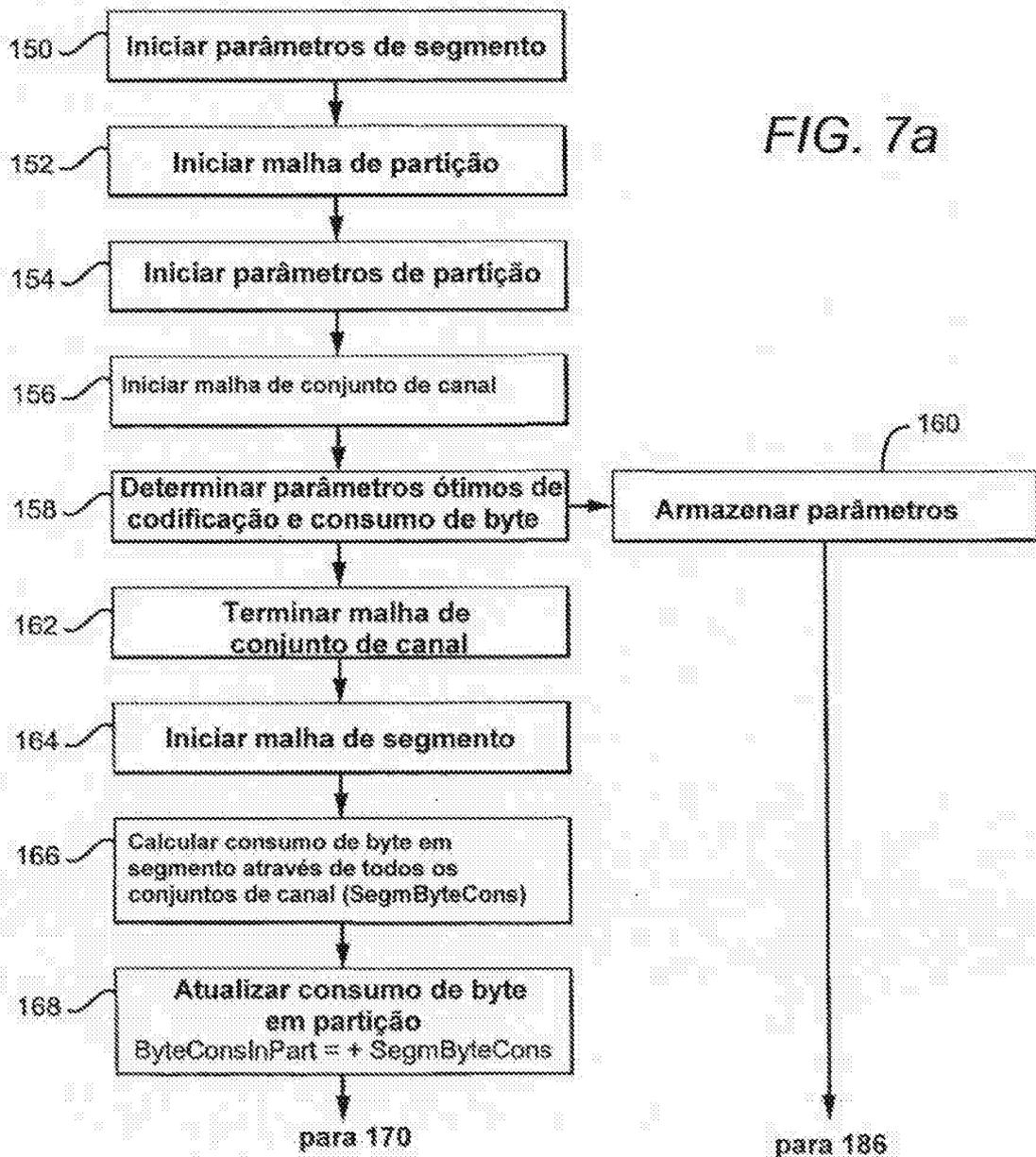


FIG. 6a

FIG. 6b





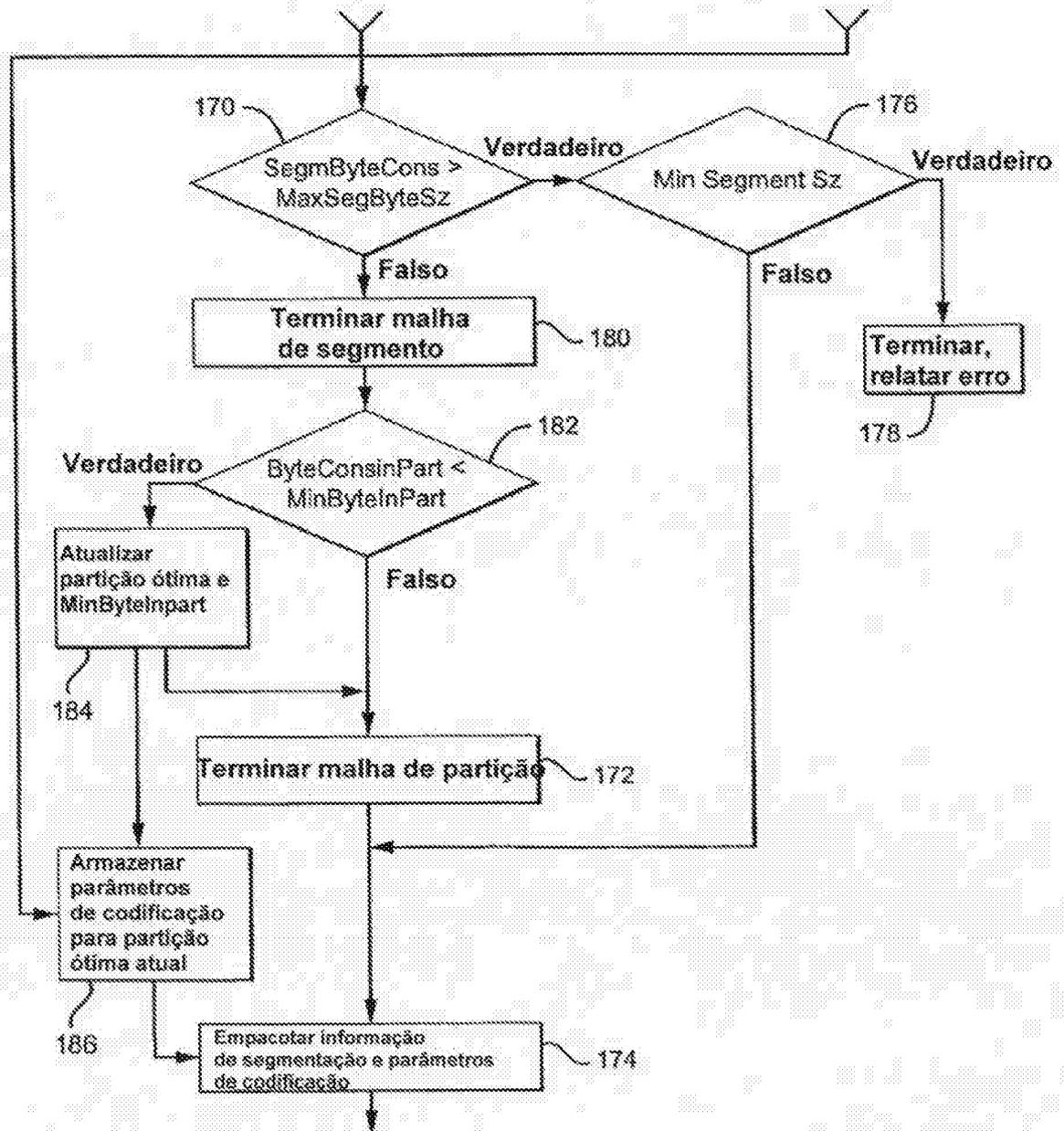
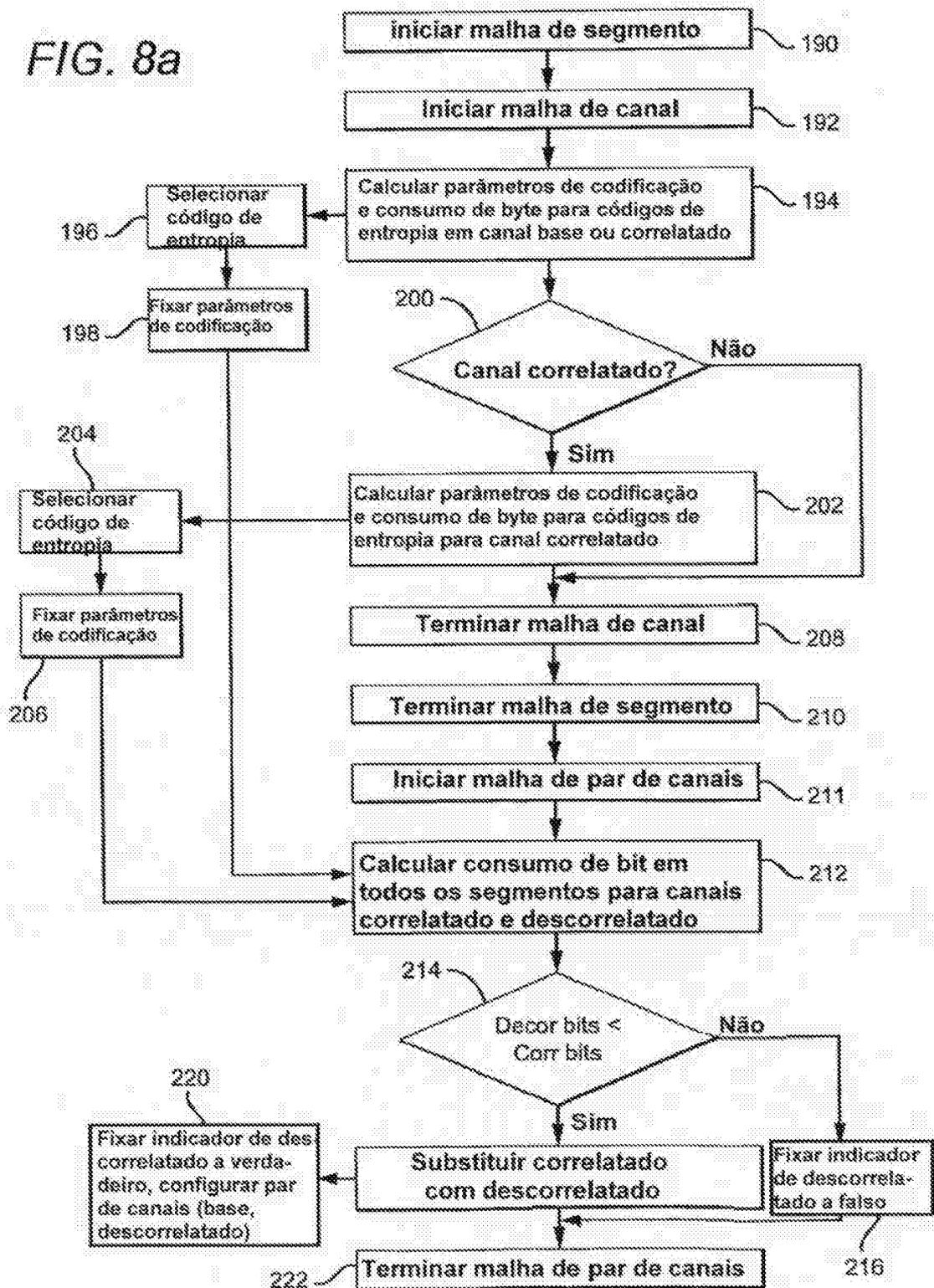


FIG. 7b

FIG. 8a



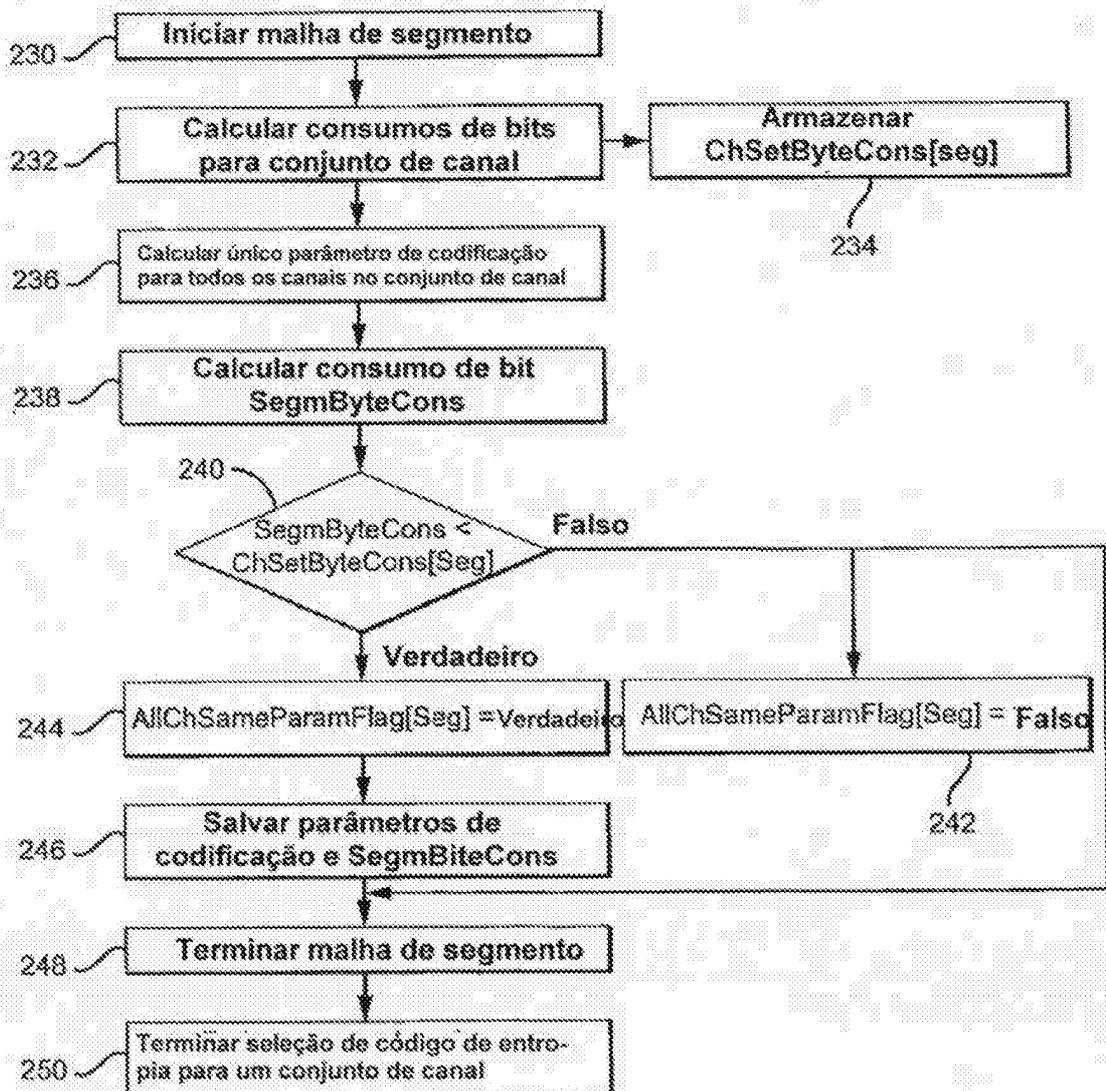
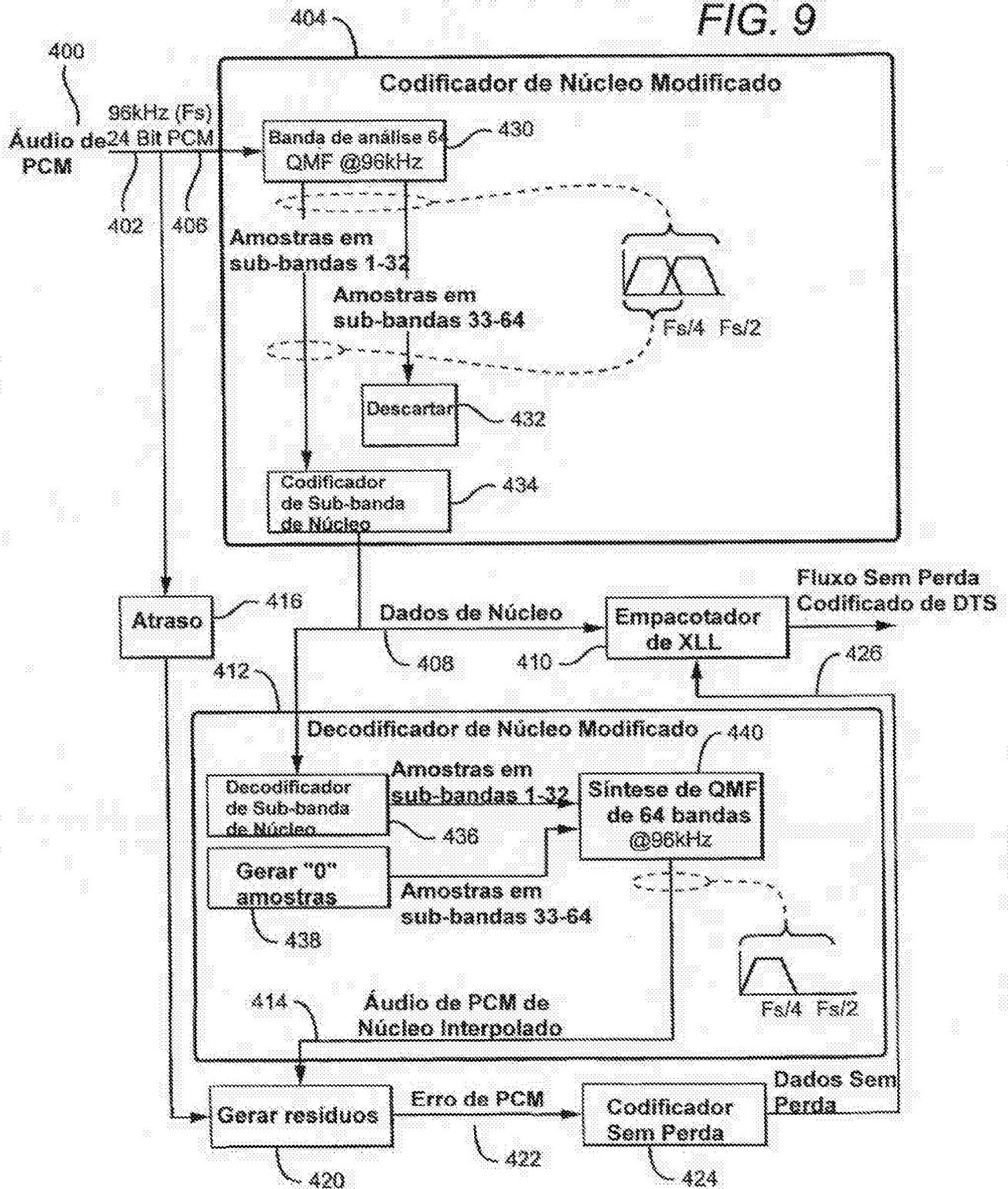


FIG. 8b

FIG. 9



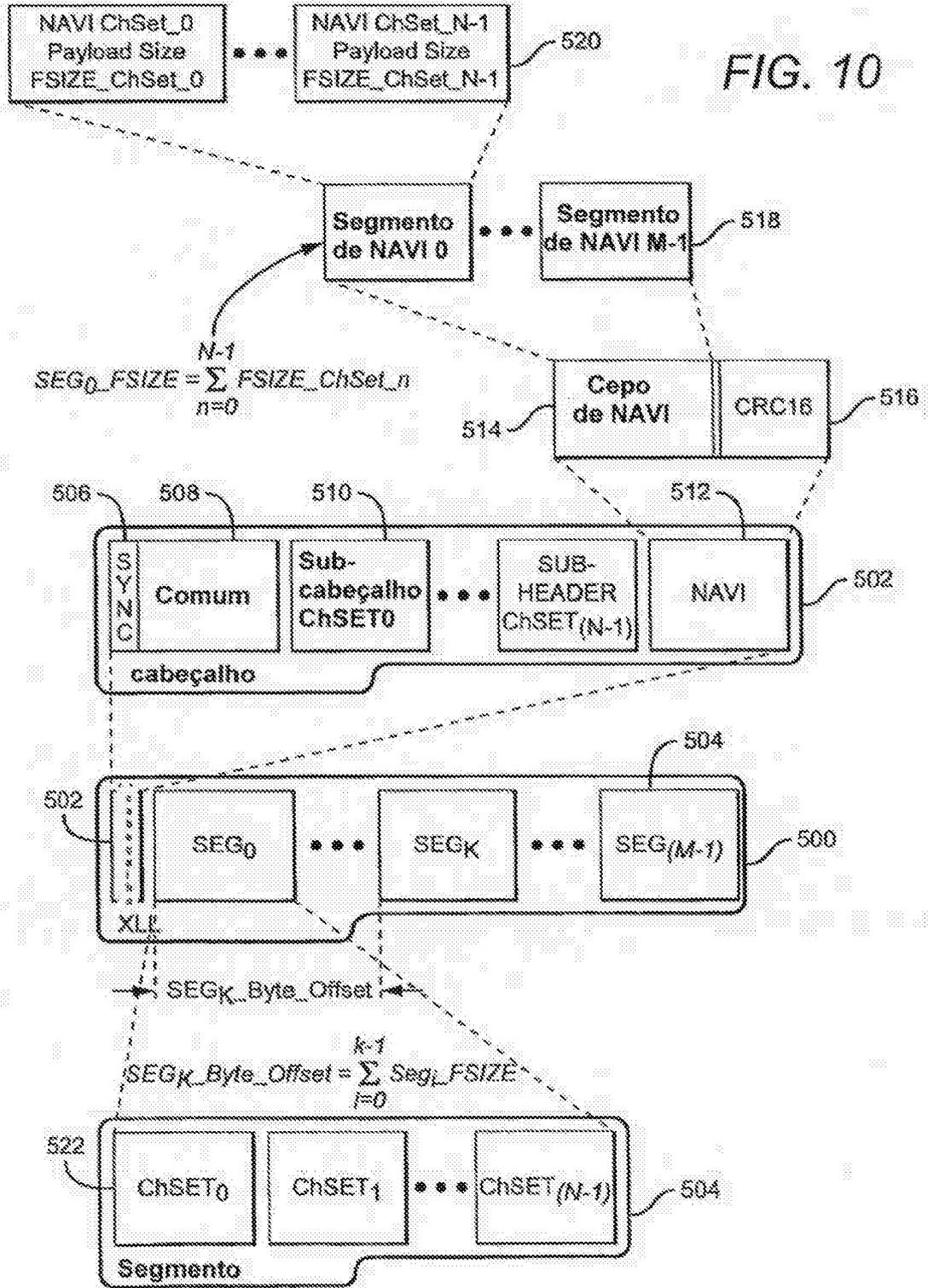


FIG. 10

FIG. 11a

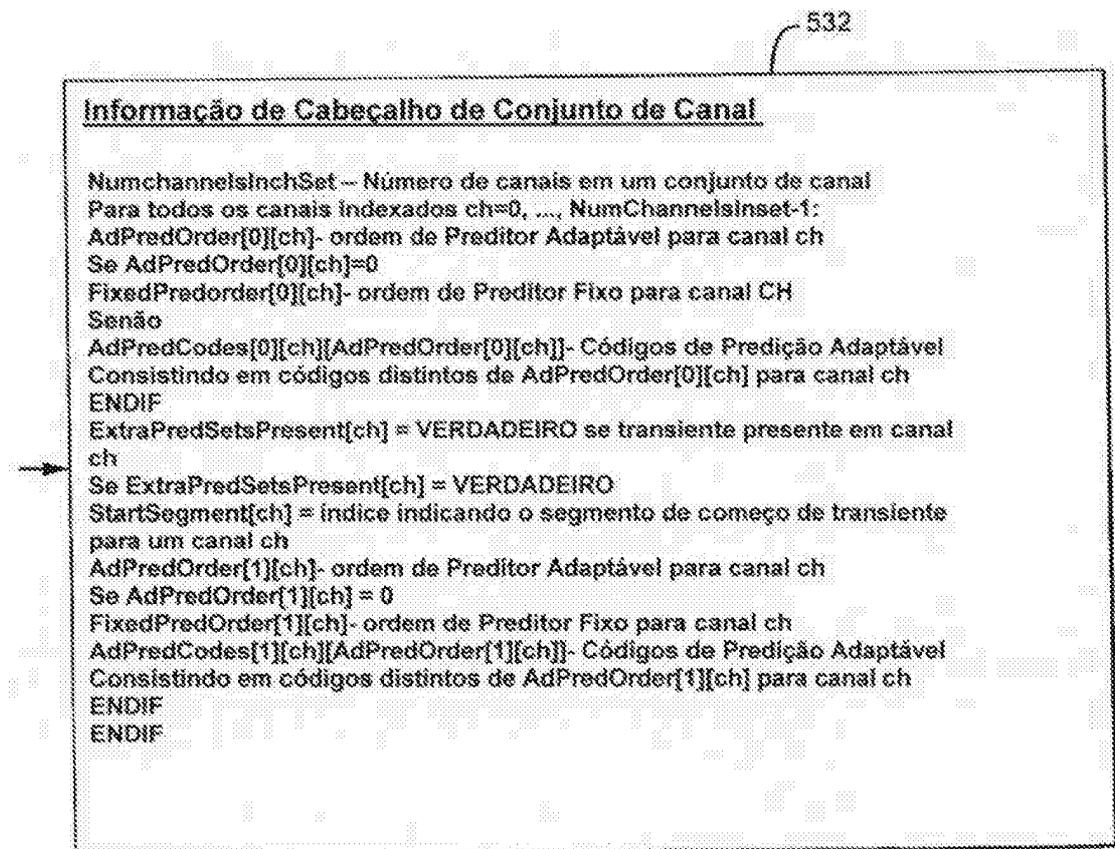
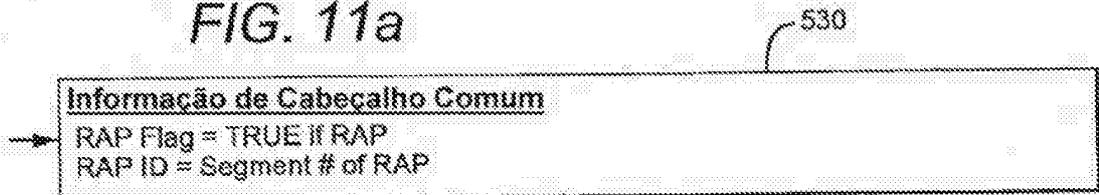
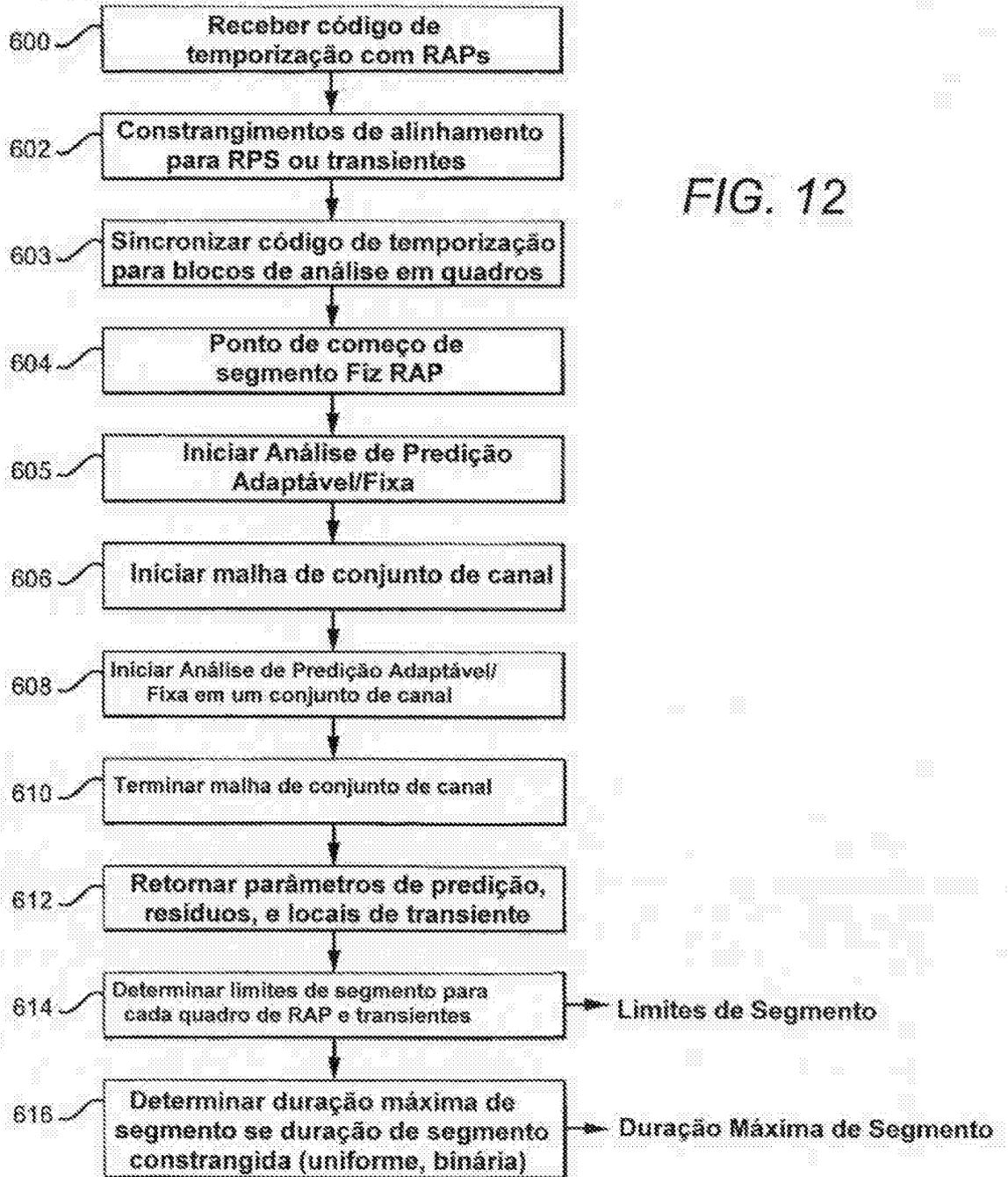


FIG. 11b



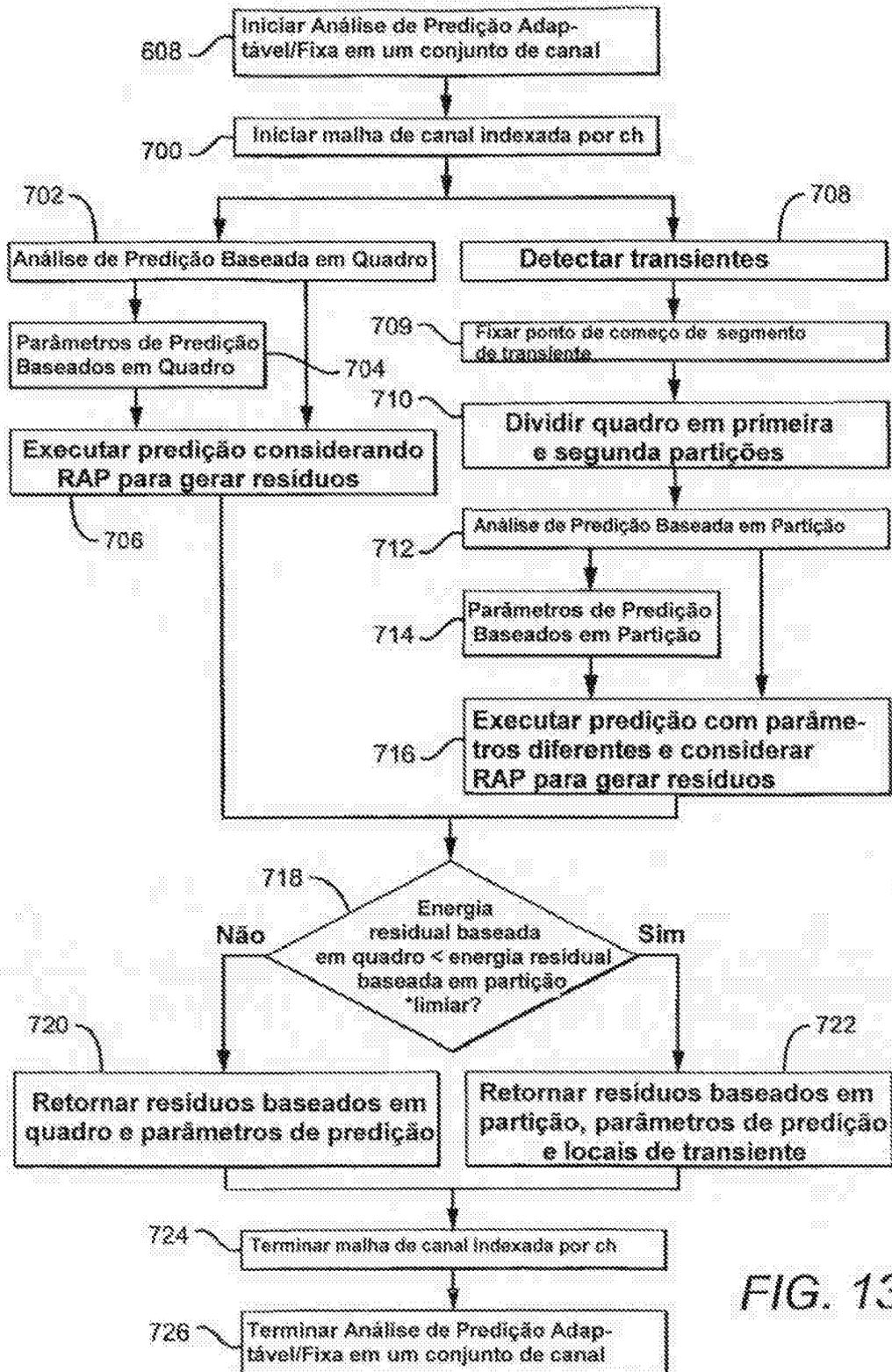
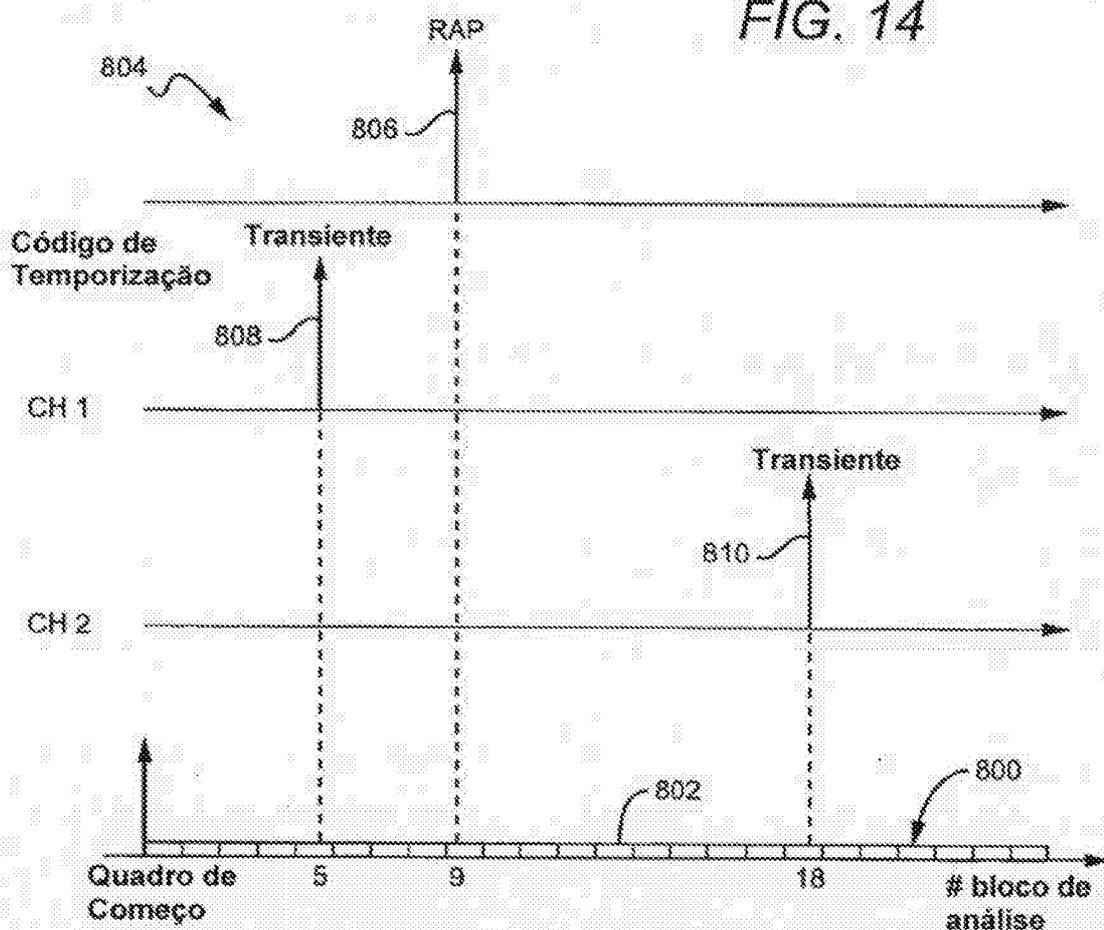


FIG. 13

FIG. 14



Duração de Segmento Constrangida

Duração Máxima de RAP = 8 ABs

CH 1 max = 4 ABs

CH 2 max = 1 AB, if 2nd AB OK than 16

Duração Máxima de Quadro de Pico = 4 x AB_dur.

Duração de Segmento não Constrangida

Início de Segmento 1: 1º segmento

Início de Segmento 2: 5º segmento

Início de Segmento 3: 8º segmento

Início de Segmento 4: 18º segmento

FIG. 15a

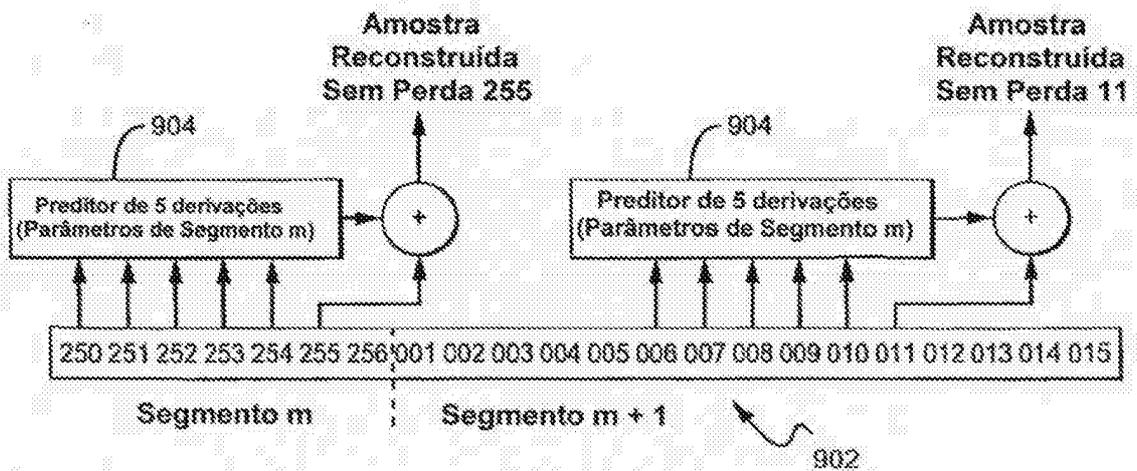
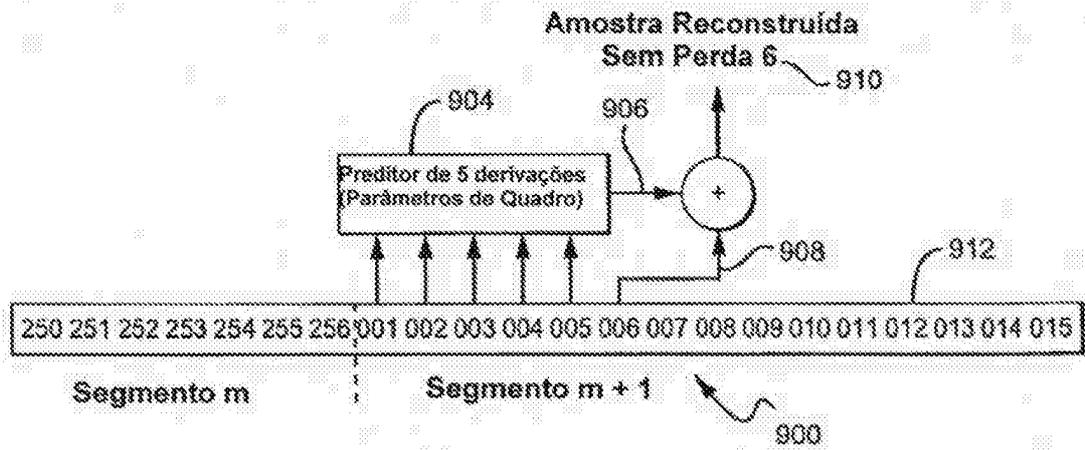


FIG. 15b

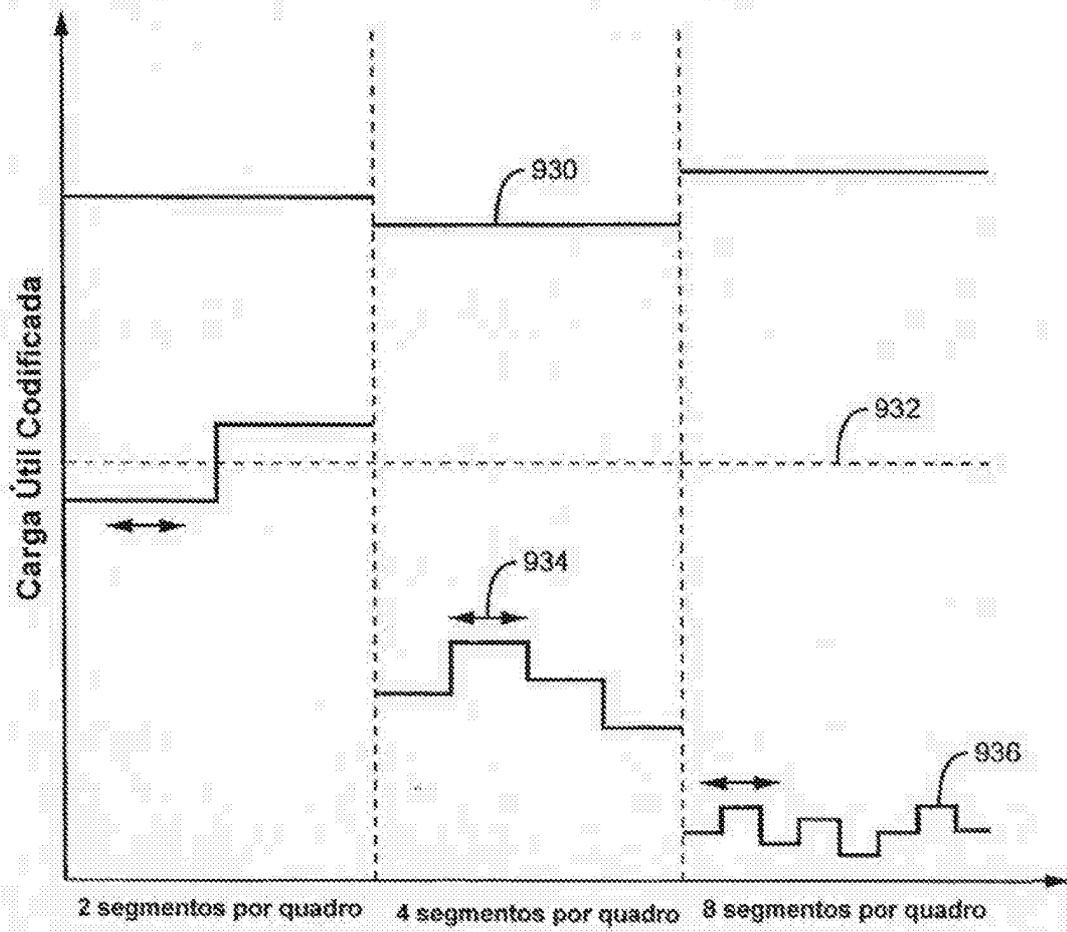


FIG. 16

RESUMO

“MÉTODOS PARA CODIFICAR ÁUDIO DE MULTICANAL, PARA
DECODIFICAR UM FLUXO DE ÁUDIO DE MULTICANAL DE TAXA
DE BIT VARIÁVEL SEM PERDA E PARA INICIAR DECODIFICAÇÃO
5 DE UM FLUXO DE BIT DE ÁUDIO DE MULTICANAL DE TAXA DE
BIT VARIÁVEL, UM OU MAIS MEIOS LEGÍVEIS POR COMPUTADOR,
UM OU MAIS DISPOSITIVOS SEMICONDUTORES, E,
DECODIFICADOR DE ÁUDIO DE MULTICANAL.”

Um codec de áudio sem perda codifica/decodifica um fluxo de
10 bits de taxa de bit variável (VBR) sem perda com capacidade de ponto de
acesso aleatório (RAP) para iniciar decodificação sem perda a um segmento
especificado dentro de um quadro e/ou capacidade de múltiplos conjuntos de
parâmetro de predição (MPPS) dividido para diminuir efeitos transientes. Isto
é realizado com uma técnica de segmentação adaptável que fixa pontos de
15 começo de segmento baseado em constrangimentos impostos pela existência
de um RAP desejado e/ou transiente detectado no quadro e seleciona uma
duração ótima de segmento em cada quadro para reduzir carga útil de quadro
codificado sujeito a um constrangimento de carga útil de segmento codificado
RAP e MPPS são particularmente aplicáveis para melhorar desempenho
20 global para durações de quadro mais longas.