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

(21) BR 112019024653-3 A2

(22) Data do Depósito: 24/05/2018

(43) Data da Publicação Nacional: 09/06/2020

(54) Título: SINALIZAÇÃO DE ALTO NÍVEL PARA DADOS DE VÍDEO FISHEYE

(51) Int. Cl.: H04N 21/235; H04N 21/81; H04N 21/854; H04N 21/236; H04N 21/845.

(30) Prioridade Unionista: 23/05/2018 US 15/987,231; 25/05/2017 US 62/511,189.

(71) Depositante(es): QUALCOMM INCORPORATED.

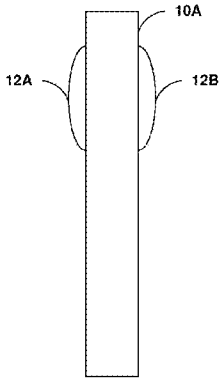
(72) Inventor(es): YE-KUI WANG; NING BI; BIJAN FORUTANPOUR.

(86) Pedido PCT: PCT US2018034435 de 24/05/2018

(87) Publicação PCT: WO 2018/218047 de 29/11/2018

(85) Data da Fase Nacional: 22/11/2019

(57) Resumo: Um exemplo de método inclui o processamento de um arquivo incluindo dados de vídeo fisheye, o arquivo incluindo uma estrutura de sintaxe incluindo uma pluralidade de elementos de sintaxe que especificam atributos dos dados de vídeo fisheye, em que a pluralidade de elementos de sintaxe inclui: um primeiro elemento de sintaxe que indica explicitamente se os dados de vídeo fisheye são monoscópicos ou estereoscópicos e um ou mais elementos de sintaxe que indicam implicitamente se os dados do vídeo fisheye são monoscópicos ou estereoscópicos; determinar, com base no primeiro elemento de sintaxe, se os dados de vídeo de fisheye são monoscópicos ou estereoscópicos; e renderizar, com base na determinação, os dados de vídeo fisheye como monoscópicos ou estereoscópicos.



"SINALIZAÇÃO DE ALTO NÍVEL PARA DADOS DE VÍDEO FISHEYE"

[001] Este pedido reivindica o benefício do Pedido de Patente provisório US número de série 62/511,189, depositado em 25 de maio de 2017, cujo conteúdo integral está aqui incorporado a título de referência.

CAMPO TÉCNICO

[002] Esta revelação refere-se ao armazenamento e transporte de dados de vídeo codificados.

ANTECEDENTES

[003] Capacidades de vídeo digitais podem ser incorporadas em uma ampla gama de dispositivos, incluindo televisores digitais, sistemas de transmissão digital direta, sistemas de transmissão sem fio, assistentes digitais pessoais (PDAs), laptops ou computadores de mesa, câmeras digitais, dispositivos de gravação digital, reprodutores de mídia digitais, dispositivos de jogos de vídeo, videogames, telefones celulares de rádio ou satélite, dispositivos de vídeo de teleconferência e afins. Dispositivos de vídeo digital implementam técnicas de compressão de vídeo, como aquelas descritas nos padrões definidos pelo MPEG-2, MPEG-4, ITU-T H.263 ou ITU-T H.264/MPEG-4, Parte 10, Codificação de Vídeo Avançada (AVC), ITU-T H.265 (também chamado de Codificação de Vídeo de Alta Eficiência (HEVC)) e extensões de tais padrões para transmitir e receber informações de vídeo digital de forma mais eficiente.

[004] Técnicas de compressão de vídeo realizam a previsão espacial e/ou previsão temporal para reduzir ou eliminar a redundância inerente nas sequências de vídeo. Para a codificação de vídeo baseada em bloco, um

quadro de vídeo ou fatia pode ser dividido em macroblocos. Cada macrobloco pode ser dividido adicionalmente. Os macroblocos em um quadro ou fatia intra-codificada (I) são codificados usando a previsão espacial em relação aos macroblocos vizinhos. Os macroblocos em um quadro ou fatia inter-codificada (P ou B) podem usar a previsão espacial em relação aos macroblocos vizinhos no mesmo quadro ou fatia, ou a previsão temporal, no que diz respeito a outras amostras de referência.

[005] Após a codificação dos dados de vídeo, os dados de vídeo podem ser empacotados para transmissão ou armazenamento. Os dados de vídeo podem ser reunidos em um arquivo de vídeo em conformidade com qualquer uma de várias normas, como o formato de arquivo de mídia base da Organização Internacional para Padronização (ISO) e suas extensões, como o AVC.

SUMÁRIO

[006] Em um exemplo, um método inclui o processamento de um arquivo incluindo dados de vídeo fisheye, o arquivo incluindo uma estrutura de sintaxe incluindo uma pluralidade de elementos de sintaxe que especificam atributos dos dados de vídeo fisheye, em que a pluralidade de elementos de sintaxe inclui: um primeiro elemento de sintaxe que indica explicitamente se os dados de vídeo fisheye são monoscópicos ou estereoscópicos e um ou mais elementos de sintaxe que indicam implicitamente se os dados do vídeo fisheye são monoscópicos ou estereoscópicos; determinar, com base no primeiro elemento de sintaxe, se os dados de vídeo de fisheye são monoscópicos ou estereoscópicos; e renderizar, com base na

determinação, os dados de vídeo fisheye como monoscópicos ou estereoscópicos.

[007] Em um outro exemplo, um dispositivo inclui uma memória configurada para armazenar pelo menos uma porção de um arquivo incluindo dados de vídeo fisheye, o arquivo incluindo uma estrutura de sintaxe incluindo uma pluralidade de elementos de sintaxe que especificam atributos dos dados de vídeo fisheye, em que a pluralidade de elementos de sintaxe inclui: um primeiro elemento de sintaxe que indica explicitamente se os dados de vídeo fisheye são monoscópicos ou estereoscópicos e um ou mais elementos de sintaxe que indicam implicitamente se os dados do vídeo fisheye são monoscópicos ou estereoscópicos; e um ou mais processadores configurados para: determinar, com base no primeiro elemento de sintaxe, se os dados de vídeo de fisheye são monoscópicos ou estereoscópicos; e renderizar, com base na determinação, os dados de vídeo fisheye como monoscópicos ou estereoscópicos.

[008] Em outro exemplo, um método inclui obter dados de vídeo fisheye e parâmetros extrínsecos de câmeras usadas para capturar os dados de vídeo fisheye; determinar, com base nos parâmetros extrínsecos, se os dados de vídeo fisheye são monoscópicos ou estereoscópicos; e codificar, em um arquivo, os dados de vídeo fisheye e uma estrutura de sintaxe, incluindo uma pluralidade de elementos de sintaxe que especificam atributos dos dados de vídeo fisheye, em que a pluralidade de elementos de sintaxe inclui: um primeiro elemento de sintaxe que indica explicitamente se os dados de vídeo fisheye são monoscópicos ou estereoscópicos e um ou mais elementos de

sintaxe que indicam explicitamente os parâmetros extrínsecos das câmeras usadas para capturar os dados de vídeo fisheye.

[009] Em outro exemplo, um dispositivo inclui uma memória configurada para armazenar nos dados de vídeo fisheye; e um ou mais processadores configurados para: obter parâmetros extrínsecos de câmeras usadas para capturar os dados de vídeo fisheye; determinar, com base nos parâmetros extrínsecos, se os dados de vídeo fisheye são monoscópicos ou estereoscópicos; e codificar, em um arquivo, os dados de vídeo fisheye e uma estrutura de sintaxe, incluindo uma pluralidade de elementos de sintaxe que especificam atributos dos dados de vídeo fisheye, em que a pluralidade de elementos de sintaxe inclui: um primeiro elemento de sintaxe que indica explicitamente se os dados de vídeo fisheye são monoscópicos ou estereoscópicos e um ou mais elementos de sintaxe que indicam explicitamente os parâmetros extrínsecos das câmeras usadas para capturar os dados de vídeo fisheye.

[0010] Os detalhes de um ou mais exemplos são apresentados nos desenhos de acompanhamento e na descrição abaixo. Outras características, objetos e vantagens serão evidentes a partir da descrição, desenhos e reivindicações.

BREVE DESCRIÇÃO DOS DESENHOS

[0011] As FIGs. 1A e 1B são diagramas de blocos que ilustram exemplos de dispositivos para capturar conteúdo de imagem omnidirecional, de acordo com uma ou mais técnicas exemplares descritas nesta revelação.

[0012] A FIG. 2 é um exemplo de uma imagem que inclui várias imagens fisheye.

[0013] As FIGs. 3-8 são diagramas conceituais que ilustram vários parâmetros extrínsecos e campos de visão de imagens omnidirecionais, de acordo com um ou mais exemplos de técnicas descritas nesta revelação.

[0014] A FIG. 9 é um diagrama de blocos que ilustra um sistema exemplar que implementa técnicas para fluxo contínuo de dados de mídia sobre uma rede.

[0015] A FIG. 10 é um diagrama conceitual ilustrando elementos do conteúdo multimídia exemplar.

[0016] A FIG. 11 é um diagrama de blocos que ilustra elementos de um arquivo de vídeo de exemplo, que pode corresponder a um segmento de uma representação.

[0017] A FIG. 12 é um fluxograma que ilustra uma técnica de exemplo para processar um arquivo que inclui dados de vídeo fisheye, de acordo com uma ou mais técnicas desta revelação.

[0018] A FIG. 13 é um fluxograma que ilustra uma técnica de exemplo para gerar um arquivo que inclui dados de vídeo fisheye, de acordo com uma ou mais técnicas desta revelação.

DESCRIÇÃO DETALHADA

[0019] As técnicas de exemplo descritas nesta revelação estão relacionadas ao processamento de arquivos que representam dados de vídeo ou imagem omnidirecionais. Quando o conteúdo de mídia omnidirecional é consumido com certos dispositivos (por exemplo, um monitor montado na cabeça e fones de ouvido), apenas as partes da mídia que correspondem à orientação de visualização do usuário são renderizadas, como se o usuário estivesse no local onde e quando a mídia foi capturada (por exemplo, onde estavam as

câmeras). Uma das formas mais populares de aplicativos de mídia omnidirecional é o vídeo omnidirecional, também conhecido como vídeo em 360 graus. O vídeo omnidirecional é normalmente capturado por várias câmeras que cobrem até 360 graus da cena.

[0020] Em geral, o vídeo omnidirecional é formado a partir de uma sequência de imagens omnidirecionais. Por conseguinte, as técnicas de exemplo descritas nesta revelação são descritas com relação à geração de conteúdo de imagem omnidirecional. Em seguida, para conteúdo de vídeo omnidirecional, essas imagens omnidirecionais podem ser exibidas sequencialmente. Em alguns exemplos, um usuário pode desejar tirar apenas uma imagem omnidirecional (por exemplo, como uma captura instantânea de todo o ambiente em 360 graus do usuário), e as técnicas descritas nesta revelação também são aplicáveis a esses casos de exemplo.

[0021] O vídeo omnidirecional pode ser estereoscópico ou monoscópico. Quando o vídeo é estereoscópico, uma imagem diferente é exibida para cada olho, de modo que o espectador perceba a profundidade. Como tal, o vídeo estereoscópico é normalmente capturado usando duas câmeras voltadas para cada direção. Quando o vídeo é monoscópico, a mesma imagem é mostrada para os dois olhos.

[0022] Os dados de vídeo podem ser considerados dados de vídeo fisheye, onde são capturados usando uma ou mais lentes fisheye (ou gerados para parecer que foram capturados usando uma ou mais lentes fisheye). Uma lente fisheye pode ser uma lente de grande angular que produz uma forte distorção visual destinada a criar uma

imagem panorâmica ou hemisférica ampla.

[0023] As técnicas podem ser aplicáveis ao conteúdo de vídeo capturado, realidade virtual e, geralmente, à exibição de vídeo e imagem. As técnicas podem ser usadas em dispositivos móveis, mas as técnicas não devem ser consideradas limitadas a aplicativos móveis. Em geral, as técnicas podem ser para aplicativos de realidade virtual, aplicativos de videogame ou outros aplicativos em que um ambiente de imagem/vídeo esférico de 360 graus é desejado.

[0024] Em alguns exemplos, o conteúdo da imagem omnidirecional pode ser capturado com um dispositivo de câmera que inclui duas lentes fisheye. Onde as duas lentes fisheye estão posicionadas em lados opostos do dispositivo da câmera para capturar partes opostas da esfera do conteúdo da imagem, o conteúdo da imagem pode ser monoscópico e cobrir toda a esfera do vídeo em 360 graus. Da mesma forma, onde as duas lentes fisheye estão posicionadas no mesmo lado do dispositivo da câmera para capturar a mesma parte da esfera do conteúdo da imagem, o conteúdo da imagem pode ser estereoscópico e cobrir metade da esfera do vídeo em 360 graus. As imagens geradas pelas câmeras são imagens circulares (por exemplo, um quadro de imagem inclui duas imagens circulares).

[0025] As FIGs. 1A e 1B são diagramas de blocos que ilustram exemplos de dispositivos para capturar conteúdo de imagem omnidirecional, de acordo com uma ou mais técnicas exemplares descritas nesta revelação. Conforme ilustrado na FIG. 1A, o dispositivo de computação 10A é um dispositivo de captura de vídeo que inclui a lente

fisheye 12A e a lente fisheye 12B localizadas em lados opostos do dispositivo de computação 10A para capturar o conteúdo da imagem monoscópica que cobre toda a esfera (por exemplo, conteúdo de vídeo completo em 360 graus). Conforme ilustrado na FIG. 1B, o dispositivo de computação 10B é um dispositivo de captura de vídeo que inclui a lente fisheye 12C e a lente fisheye 12D localizadas no mesmo lado do dispositivo de computação 10B para o conteúdo da imagem estereoscópica que cobre cerca de metade da esfera.

[0026] Como descrito acima, um dispositivo de câmera inclui uma pluralidade de lentes fisheye. Alguns exemplos de dispositivos de câmera incluem duas lentes fisheye, mas as técnicas de exemplo não se limitam a duas lentes fisheye. Um exemplo de dispositivo de câmera pode incluir 16 lentes (por exemplo, conjunto de 16 câmeras para filmar conteúdo VR 3D). Outro exemplo de dispositivo de câmera pode incluir oito lentes, cada uma com um ângulo de visão de 195 graus (por exemplo, cada lente captura 195 graus dos 360 graus do conteúdo da imagem). Outro exemplo de dispositivo de câmera inclui três ou quatro lentes. Alguns exemplos podem incluir uma lente de 360 graus que captura 360 graus de conteúdo da imagem.

[0027] As técnicas de exemplo descritas nesta revelação são geralmente descritas com relação a duas lentes fisheye que capturam imagem/vídeo omnidirecional. No entanto, as técnicas de exemplo não são limitadas a isso. As técnicas do exemplo podem ser aplicáveis aos dispositivos de câmera do exemplo que incluem uma pluralidade de lentes (por exemplo, duas ou mais), mesmo que as lentes não sejam lentes fisheye e uma pluralidade de

lentes fisheye. Por exemplo, as técnicas do exemplo descrevem maneiras de costurar imagens capturadas, e as técnicas podem ser aplicáveis aos exemplos em que há uma pluralidade de imagens capturadas de uma pluralidade de lentes (que podem ser lentes fisheye, como um exemplo). Embora as técnicas do exemplo sejam descritas em relação a duas lentes fisheye, as técnicas do exemplo não são limitadas a isso e são aplicáveis aos vários tipos de câmera usados para capturar imagens/vídeos omnidirecionais.

[0028] As técnicas desta revelação podem ser aplicadas aos arquivos de vídeo em conformidade com os dados de vídeo encapsulados de acordo com qualquer formato de arquivo de mídia base ISO (por exemplo, ISOBMFF, ISO/IEC 14496-12) e outros formatos de arquivo derivados do ISOBMFF, incluindo formato de arquivo MPEG-4 (ISO/IEC 14496-15), formato de arquivo do Projeto de Parceria para a Terceira Geração (3 GPP) (3 GPP TS 26.244) e formatos de arquivo para famílias AVC e HEVC dos codecs de vídeo (ISO/IEC 14496-15) ou outros formatos de arquivo de vídeo semelhantes.

[0029] O ISOBMFF é usado como base para muitos formatos de encapsulamento de codec, como o formato de arquivo AVC, bem como para muitos formatos de contêiner multimídia, como o formato de arquivo MPEG-4, o formato de arquivo 3 GPP (3GP), e o formato de arquivo DVB. Além da mídia contínua, como áudio e vídeo, a mídia estática, como imagens, bem como os metadados, pode ser armazenada em um arquivo em conformidade com o ISOBMFF. Os arquivos estruturados de acordo com o ISOBMFF podem ser usados para vários propósitos, incluindo reprodução de arquivo de mídia

local, download progressivo de um arquivo remoto, segmentos para Streaming Adaptativo Dinâmico sobre HTTP (DASH), contêineres para conteúdo a ser transmitido e suas instruções de empacotamento e gravação de fluxos de mídia recebidos em tempo real.

[0030] Uma caixa é a estrutura de sintaxe elementar no ISOBMFF, incluindo um tipo de caixa codificado com quatro caracteres, a contagem de bytes da caixa e a carga útil. Um arquivo ISOBMFF consiste em uma sequência de caixas, e as caixas podem conter outras caixas. Uma caixa de filme ("moov") contém os metadados para os fluxos de mídia contínuos presentes no arquivo, cada um representado no arquivo como uma faixa. Os metadados de uma faixa são colocados em uma caixa Faixa ("trak"), enquanto o conteúdo de mídia de uma faixa é ou colocado em uma caixa Dados de Mídia ("mdat") ou diretamente em um arquivo separado. O conteúdo da mídia das faixas consiste em uma sequência de amostras, como unidades de acesso de áudio ou vídeo.

[0031] Ao renderizar dados de vídeo fisheye, pode ser desejável que um decodificador de vídeo determine se os dados de vídeo fisheye são monoscópicos ou estereoscópicos. Por exemplo, a maneira pela qual o decodificador de vídeo exibe e/ou une as imagens circulares incluídas em uma imagem dos dados de vídeo fisheye depende diretamente se os dados de vídeo fisheye são monoscópicos ou estereoscópicos.

[0032] Em alguns exemplos, um decodificador de vídeo pode determinar se os dados de vídeo fisheye são monoscópicos ou estereoscópicos com base em um ou mais elementos de sintaxe incluídos em um arquivo que indicam

implicitamente se os dados de vídeo fisheye descritos pelo arquivo são monoscópicos ou estereoscópicos. Por exemplo, um codificador de vídeo pode incluir elementos de sintaxe no arquivo que descrevem parâmetros extrínsecos (por exemplo, atributos físicos/locais, como ângulo de guinada, ângulo de inclinação, ângulo de rotação e um ou mais deslocamentos espaciais) de cada uma das câmeras usadas para capturar os dados de vídeo fisheye e o decodificador de vídeo pode processar os parâmetros extrínsecos para calcular se os dados de vídeo descritos pelo arquivo são monoscópicos ou estereoscópicos. Como exemplo, se o processamento dos parâmetros extrínsecos levar a uma indicação de que as câmeras estão voltadas para a mesma direção e têm um deslocamento espacial, o decodificador de vídeo pode determinar que os dados de vídeo são estereoscópicos. Como outro exemplo, se o processamento dos parâmetros extrínsecos indica que as câmeras estão voltadas para direções opostas, o decodificador de vídeo pode determinar que os dados de vídeo são monoscópicos. No entanto, em alguns exemplos, pode ser desejável que o decodificador de vídeo seja capaz de determinar se os dados de vídeo fisheye são monoscópicos ou estereoscópicos sem precisar executar esses cálculos adicionais, que exigem muitos recursos. Adicionalmente, em alguns exemplos, o processamento dos parâmetros extrínsecos pode não produzir a classificação correta (por exemplo, conforme pretendida) dos dados de vídeo como monoscópicos ou estereoscópicos. Como um exemplo, os valores dos parâmetros extrínsecos podem ser corrompidos durante a codificação/trânsito/decodificação. Como outro exemplo, os

parâmetros extrínsecos podem não ter sido fornecidos com precisão no codificador.

[0033] De acordo com uma ou mais técnicas desta revelação, um codificador de vídeo pode incluir um primeiro elemento de sintaxe em um arquivo que indica explicitamente se os dados de vídeo fisheye descritos pelo arquivo são monoscópicos ou estereoscópicos. O arquivo pode incluir o primeiro elemento de sintaxe, além dos elementos de sintaxe que os parâmetros extrínsecos dos dados de vídeo fisheye. Dessa forma, o arquivo pode incluir uma indicação explícita (isto é, o primeiro elemento de sintaxe) e uma indicação implícita (isto é, os elementos de sintaxe que os parâmetros extrínsecos) de se os dados de vídeo fisheye descritos pelo arquivo são monoscópicos ou estereoscópicos. Dessa maneira, um decodificador de vídeo pode determinar com precisão se os dados de vídeo são monoscópicos ou estereoscópicos (por exemplo, sem a necessidade de realizar cálculos adicionais com base nos parâmetros extrínsecos).

[0034] O ISOBMFF especifica os seguintes tipos de faixas: uma faixa de mídia, que contém um fluxo de mídia elementar, uma faixa de dicas, que inclui instruções de transmissão de mídia ou representa um fluxo de pacotes recebido e uma faixa de metadados temporizada, que inclui metadados sincronizados no tempo.

[0035] Embora originalmente projetado para armazenamento, o ISOBMFF provou ser muito valioso para streaming, por exemplo, para download progressivo ou DASH. Para fins de streaming, os fragmentos de filme definidos em ISOBMFF podem ser usados.

[0036] Os metadados para cada faixa incluem

uma lista de entradas de descrição de amostra, cada uma fornecendo o formato de codificação ou encapsulamento usado na faixa e os dados de inicialização necessários para processar esse formato. Cada amostra é associada a uma das entradas de descrição da amostra da faixa.

[0037] O ISOBMFF permite especificar metadados específicos da amostra com vários mecanismos. Caixas específicas dentro da caixa Tabela de Amostras ("stbl") foram padronizadas para responder a necessidades comuns. Por exemplo, uma caixa de Sinc Amostra ("stss") é usada para listar as amostras de acesso aleatório da faixa. O mecanismo de agrupamento de amostras permite o mapeamento de amostras de acordo com um tipo de agrupamento de quatro caracteres em grupos de amostras que compartilham a mesma propriedade especificada como uma entrada de descrição do grupo de amostras no arquivo. Vários tipos de agrupamento foram especificados no ISOBMFF.

[0038] A realidade virtual (VR) é a capacidade de estar virtualmente presente em um mundo não físico criado pela renderização de imagens e sons naturais e/ou sintéticos correlacionados pelos movimentos do usuário imersos, permitindo interagir com esse mundo. Com o recente progresso alcançado em dispositivos de renderização, como monitores montados na cabeça (FDVID) e criação de vídeo VR (geralmente também conhecido como vídeo em 360 graus), uma qualidade significativa da experiência pode ser oferecida. Aplicativos de VR, incluindo jogos, treinamento, educação, vídeo esportivo, compras online, entretenimento adulto e assim por diante.

[0039] Um sistema VR típico pode incluir um ou

mais dos seguintes componentes, que podem executar uma ou mais das seguintes etapas:

1) Um conjunto de câmeras, que normalmente consiste em várias câmeras individuais apontando para direções diferentes e, idealmente, cobrindo coletivamente todos os pontos de vista ao redor do conjunto de câmeras.
2) Costura de imagem, em que as imagens de vídeo tiradas por várias câmeras individuais são sincronizadas no domínio do tempo e costuradas no domínio do espaço, para ser um vídeo esférico, mas mapeado para um formato retangular, como equi-retangular (como um mapa-múndi)) ou mapa do cubo.
3) O vídeo no formato retangular mapeado é codificado/compactado usando um codec de vídeo, por exemplo, H.265/HEVC ou H.264/AVC.
4) O fluxo de bits de vídeo compactado pode ser armazenado e/ou encapsulado em um formato de mídia e transmitido (possivelmente apenas o subconjunto que cobre apenas a área sendo vista por um usuário) através de uma rede para um receptor.
5) O receptor recebe o bitstream(s) de vídeo ou parte dele, possivelmente encapsulado em um formato, e envia o sinal de vídeo decodificado ou parte dele para um dispositivo de renderização.
6) O dispositivo de renderização pode ser, por

exemplo, um FDVID, que pode rastrear o movimento da cabeça e até o momento do movimento dos olhos e renderizar a parte correspondente do vídeo, de modo que uma experiência imersiva seja entregue ao usuário.

[0040] O Omnidirecional Media Application Format (OMAF) está sendo desenvolvido pelo MPEG para definir um formato de aplicativo de mídia que permita aplicativos de mídia omnidirecional, com foco em aplicativos de VR com vídeo em 360° e áudio associados. O OMAF especifica a projeção e empacotamento por região que podem ser usados para a conversão de um vídeo esférico ou de 360 ° em um vídeo retangular bidimensional, seguido de como armazenar mídia omnidirecional e os metadados associados usando o formato de arquivo de mídia base ISO (ISOBMFF) e como encapsular, sinalizar e transmitir mídia omnidirecional usando streaming adaptativo dinâmico por HTTP (DASH) e, finalmente, quais codecs de vídeo e áudio, bem como configurações de codificação de mídia, podem ser usados para compactação e reprodução do sinal de mídia omnidirecional.

[0041] Projeção e empacotamento por região são os processos usados no lado da produção de conteúdo para gerar imagens de vídeo em 2D a partir do sinal de esfera para o vídeo omnidirecional projetado. A projeção geralmente acompanha a costura, o que pode gerar o sinal de esfera a partir de várias imagens capturadas pela câmera para cada imagem de vídeo. A projeção pode ser uma etapa fundamental do processamento no vídeo VR. Os tipos típicos

de projeção incluem equi-retangular e cubemap. O Padrão Internacional de Rascunho OMAF (DIS) suporta apenas o tipo de projeção equi-retangular. A embalagem por região é uma etapa opcional após a projeção (na janela de exibição do lado da produção de conteúdo). A embalagem por região permite manipulações (redimensionamento, reposicionamento, rotação e espelhamento) de qualquer região retangular da imagem compactada antes da codificação.

[0042] A FIG. 2 é um exemplo de uma imagem que inclui várias imagens fisheye. O OMAF DIS suporta um formato de vídeo fisheye VR/360, em que, em vez de aplicar uma projeção e, opcionalmente, um empacotamento por região para gerar o vídeo 2D antes da codificação, para cada unidade de acesso, as imagens circulares das câmeras de captura são incorporadas diretamente em uma imageado 2D. Por exemplo, como mostrado na FIG. 2, a primeira imagem fisheye 202 e segunda imagem fisheye 204 e incorporada na imagem 2D 200.

[0043] Esse vídeo fisheye pode então ser codificado e o fluxo de bits pode ser encapsulado em um arquivo ISOBMFF e pode ser ainda mais encapsulado como uma representação DASH. Além disso, a propriedade do vídeo fisheye, incluindo parâmetros indicando as características do vídeo fisheye, pode ser sinalizada e usada para renderizar corretamente o video 360 no lado do cliente. Uma vantagem da abordagem de vídeo fisheye VR/360 é que ele suporta conteúdo de VR de baixo custo gerado por usuários por terminais móveis.

[0044] No OMAF DIS, o uso do esquema de vídeo omnidirecional fisheye para o tipo de entrada de amostra de

vídeo restrito 'resv' indica que as imagens decodificadas são imagens de vídeo fisheye. O uso do esquema de vídeo omnidirecional fisheye é indicado pelo tipo de esquema igual a 'fodv' (vídeo omnidirecional fisheye) na SchemeTypeBox. O formato do vídeo fisheye é indicado com o FisheyeOmnidirectionalVideoBox contido no SchemeInformationBox, que é incluído no RestnctedSchemeInfoBox que está incluído na entrada da amostra. No rascunho atual do OMAF DIS (Tecnologia da Informação - Representação codificada de mídia imersiva (MPEG-I) - Parte 2: formato de mídia omnidirecional, ISO/IEC FDIS 14496-15: 2014 (E), ISO/IEC JTC 1/SC 29/WG 11, w16824, 2014-01-13, a seguir "rascunho atual do OMAF DIS"), uma e apenas uma FisheyeOmnidirectionalVideoBox deve estar presente na SchemeInformationBox quando o tipo de esquema for 'fodv'. Quando a FisheyeOmnidirectionalVideoBox estiver presente no SchemeInformationBox, a StereoVideoBox e a RegionWisePackingBox não estarão presentes na mesma SchemeInformationBox. A FisheyeOmnidirectionalVideoBox, conforme especificado na cláusula 6 do OMAF DIS, contém a estrutura da sintaxe FisheyeOmnidirectionalVideoInfo( ) que contém os parâmetros de propriedade de vídeo fisheye.

[0045] A sintaxe da estrutura da sintaxe FisheyeOmnidirectionalVideoInfo( ) no rascunho atual do OMAF DIS é a seguinte.

```
aligned(8) class FisheyeOmnidirectionalVideoInfo( ) {
    bit(24) reserved = 0;
    unsigned int(8) num_circular_images;
    for(i=0; i< num_circular_images; i++) {
        unsigned int(32) image_center_x;
        unsigned int(32) image_center_y;
        unsigned int(32) full_radius;
        unsigned int(32) picture_radius;
        unsigned int(32) scene_radius;
        unsigned int(32) image_rotation;
        bit(30) reserved = 0;
        unsigned int(2) image_flip;
        unsigned int(32) image_scale_axis_angle;
        unsigned int(32) image_scale_x;
        unsigned int(32) image_scale_y;
        unsigned int(32) field_of_view;
        bit(16) reserved = 0;
        unsigned int (16) num_angle_for_displaying_fov;
        for(j=0; j< num_angle_for_displaying_fov; j++) {
```

```
        unsigned int(32) displayed_fov;
        unsigned int(32) overlapped_fov;
    }
    signed int(32) camera_center_yaw;
    signed int(32) camera_center_pitch;
    signed int(32) camera_center_roll;
    unsigned int(32) camera_center_offset_x;
    unsigned int(32) camera_center_offset_y;
    unsigned int(32) camera_center_offset_z;
    bit(16) reserved = 0;
    unsigned int(16) num_polynomial_coefficeients;
    for(j=0; j< num_polynomial_coefficients; j++) {
        unsigned int(32) polynomial_coefficient_K;
    }
    bit(16) reserved = 0;
    unsigned int (16) num_local_fov_region;
    for(j=0; j<num_local_fov_region; j++) {
        unsigned int(32) start_radius;
        unsigned int(32) end_radius;
        signed int(32) start_angle;
        signed int(32) end_angle;
        unsigned int(32) radius_delta;
        signed int(32) angle_delta;
        for(rad=start_radius; rad<= end_radius; rad+=radius_delta) {
            for(ang=start_angle; ang<= ang_radius; ang+=angle_delta) {
                unsigned int(32) local_fov_weight;
            }
        }
    }
    bit(16) reserved = 0;
    unsigned int(16) num_polynomial_coefficients_lsc;
    for(j=0; j< num_polynomial_coefficients_lsc; j++) {
        unsigned int (32) polynomial_coefficient_K_lsc_R;
        unsigned int (32) polynomial_coefficient_K_lsc_G;
```

```
                unsigned int (32) polynomial_coefficient_K_lsc_B;
            }
        }
        bit(24) reserved = 0;
        unsigned int(8) num_deadzones;
        for(i=0; i< num_deadzones; i++) {
            unsigned int(16) deadzone_left_horizontal_offset;
            unsigned int(16) deadzone_top_vertical_offset;
            unsigned int(16) deadzone_width;
            unsigned int(16) deadzone_height;
        }
    }
```

[0046] A semântica da estrutura da sintaxe FisheyeOmnidirectionalVideoInfo( ) no rascunho atual do OMAF DIS é a seguinte.

[0047] num_circular_images especifica o número de imagens circulares na imagem codificada de cada amostra à qual esta caixa se aplica. Normalmente, o valor é igual a 2, mas outros valores diferentes de zero também são possíveis.

[0048] image_center_x é um valor de ponto fixo 16.16 que especifica a coordenada horizontal, em amostras luma, do centro da imagem circular na imagem codificada de cada amostra à qual esta caixa se aplica.

[0049] image_center_y é um valor de ponto fixo 16.16 que especifica a coordenada vertical, em amostras luma, do centro da imagem circular na imagem codificada de cada amostra à qual esta caixa se aplica.

[0050] full_radius é um valor de ponto fixo

16.16 que especifica o raio, em amostras luma, do centro da imagem circular até a borda da imagem redonda completa.

[0051] picture_radius é um valor de ponto fixo 16.16 que especifica o raio, em amostras luma, do centro da imagem circular até a borda mais próxima da borda da imagem. A imagem fisheye circular pode ser cortada pela imagem da câmera. Portanto, esse valor indica o raio de um círculo em que os pixels são utilizáveis.

[0052] scene_radius é um valor de ponto fixo 16.16 que especifica o raio, em amostras luma, do centro da imagem circular até a borda mais próxima da área na imagem, onde é garantido que não há obstruções no corpo da câmera em si e que dentro da área fechada não há distorção da lente muito grande para a costura.

[0053] image_rotation é um valor de ponto fixo 16.16 que especifica a quantidade de rotação, em graus, da imagem circular. A imagem pode ser girada por imagens +/- 90 graus, ou +/- 180 graus, ou qualquer outro valor.

[0054] image_flip especifica se e como a imagem foi invertida e, portanto, uma operação de inversão reversa precisa ser aplicada. O valor 0 indica que a imagem não foi invertida. O valor 1 indica que a imagem foi invertida verticalmente. O valor 2 indica que a imagem foi invertida horizontalmente. O valor 3 indica que a imagem foi invertida tanto verticalmente quanto horizontalmente.

[0055] Image_scale_axis_angle, image_scale x e image_scale_y são três valores 16.16 de ponto fixo que especificam se e como a imagem foi dimensionada ao longo de um eixo. O eixo é definido por um único ângulo, conforme indicado pelo valor do ângulo do eixo da escala da imagem,

em graus. Um ângulo de 0 graus significa que um vetor horizontal é perfeitamente horizontal e um vetor vertical é perfeitamente vertical. Os valores da image_scale_x e image_scale_y indicam as proporções de escala nas direções paralelas e ortogonais, respectivamente, ao eixo.

[0056] field_of_view é um valor de ponto fixo 16.16 que especifica o campo de visão da lente fisheye, em graus. Um valor típico para uma lente fisheye hemisférica é 180,0 graus.

[0057] num_angle_for_displaying_fov especifica o número de ângulos. De acordo com o valor de num_angle_for_displaying_fov, vários valores de fov exibido e fov sobreposto são definidos com intervalos iguais, que começam às 12 horas e vão no sentido horário.

[0058] displayed_fov especifica o campo de visão exibido e a área de imagem correspondente de cada imagem da câmera fisheye, overlapped_fov especifica a região que inclui regiões sobrepostas, que geralmente são usadas para mesclar, em termos do campo de visão entre várias imagens circulares. Os valores de displayed_fov e overlapped_fov são menores ou iguais ao valor do campo de visão.

[0059] NOTA: O valor do campo de visão é determinado pelas propriedades físicas de cada lente fisheye, enquanto os valores de displayed_fov e overlapped_fov são determinados pela configuração de várias lentes fisheye. Por exemplo, quando o valor do num_circular_images é igual a 2 e duas lentes são localizadas simetricamente, o valor de dispayed_fov e overlapped_fov pode ser definido como 180 e 190

respectivamente, por padrão. No entanto, o valor pode ser alterado dependendo da configuração da lente e das características do conteúdo. Por exemplo, se a qualidade da costura com os valores de displayed_fov (câmera esquerda = 170 e câmera direita = 190) e os valores de overlapped_fov (câmera esquerda = 185 e câmera direita = 190) são melhores que a qualidade com os valores padrão (180 e 190) ou se a configuração física das câmeras for assimétrica, os valores desiguais de displayed_fov e overlapped_fov podem ser obtidos. Além disso, quando se trata de múltiplas (N> 2) imagens fisheye, um único valor de displayed_fov não pode especificar a área exata de cada imagem fisheye. Conforme mostrado na FIG. 6, o displayed_fov (602) varia de acordo com a direção. Para manipular múltiplas (N> 2) imagens fisheye, é introduzido o num_angle_for_displaying_fov. Por exemplo, se esse valor for igual a 12, a imagem fisheye será dividida em 12 setores em que cada ângulo de setor é de 30 graus.

[0060] camera_center_yaw especifica o ângulo de guinada, em unidades de 2^{-16} graus, do ponto em que o pixel central da imagem circular na imagem codificada de cada amostra é projetado para uma superfície esférica. Este é o primeiro dos 3 ângulos que especificam os parâmetros extrínsecos da câmera em relação aos eixos de coordenadas globais. camera_center_yaw deve estar no intervalo de $-180 * 2^{16}$ a $180 * 2^{16} - 1$, inclusive.

[0061] camera_center_pitch especifica o ângulo de guinada, em unidades de 2^{-16} graus, do ponto em que o pixel central da imagem circular na imagem codificada de cada amostra é projetado para uma superfície esférica.

Camera_center_pitch deve estar na faixa de $-90 * 2^{16}$ a $90 * 2^{16}$, inclusive.

[0062] camera_center_roll especifica o ângulo de rolo, em unidades de 2^{-16} graus, do ponto em que o pixel central da imagem circular na imagem codificada de cada amostra é projetado para uma superfície esférica. camera_center_roll deve estar na faixa de $-180 * 2^{16}$ a $180 * 2^{16}$, inclusive.

[0063] camera_center_offset_x, camera_center_offset_y e camera_center_offset_z são valores de ponto fixo 8,24 que indicam os valores de deslocamento XYZ da origem da esfera unitária em que os pixels na imagem circular na imagem codificada são projetados no camera_center_offset_x , camera_center_offset_y e camera_center_offset_z devem estar no intervalo de -1,0 a 1,0, inclusive.

[0064] Num_polynomial_coefficients é um número inteiro que especifica o número de coeficientes polinomiais presentes. A lista de coeficientes polinomiais K são valores de ponto fixo 8.24 que representam os coeficientes no polinômio que especificam a transformação do espaço fisheye em imagem plana não-armazenada.

[0065] num_local_fov_region especifica o número de regiões de ajuste local com campo de visão diferente.

[0066] start_radius, end_radius, start_angle e end_angle especificam a região para ajuste/distorção local para alterar o campo de visão real para exibição local, start_radius e end_radius são valores de ponto fixo 16.16 que especificam os valores mínimos e máximos do raio,

start_angle e end_angle especificam os valores mínimo e máximo dos ângulos que começam às 12 horas e aumentam no sentido horário, em unidades de 2^{-16} graus, start_angle e end_angle devem estar na faixa de $-180 * 2^{16}$ a $180 * 2^{16} - 1$, inclusive.

[0067] Radius_delta é um valor de ponto fixo 16.16 que especifica o valor do raio delta para representação de um campo de visão diferente para cada raio.

[0068] angle_delta especifica o valor do ângulo delta, em unidades de 2^{-16} graus, para representação de um campo de visão diferente para cada ângulo.

[0069] Local_fov_weight é um formato de ponto fixo de 8.24 que especifica o valor de ponderação para o campo de visão da posição especificada pelo start_radius, end_radius, start_angle, end_angle, o índice do ângulo i e o índice do raio j. O valor positivo do peso local fov especifica a expansão do campo de visão, enquanto o valor negativo especifica a contração do campo de visão.

[0070] num_polynomial_coefficients_lsc deve ser a ordem da aproximação polinomial da curva de sombreamento da lente.

[0071] polynomial_coefficient_K_lsc_R, polynomial_coefficient_K_lsc_G e polynomial_coefficient_K_lsc_B são formatos de ponto fixo 8,24 que especificam os parâmetros LSC para compensar o artefato de sombreamento que reduz a cor ao longo da direção radial. O peso de compensação (w) a ser multiplicado para a cor original é aproximado como uma função de curva do raio do centro da imagem usando uma

expressão polinomial. É formulado como $w = \sum_{=1} Pi- \cdot r^{1-1}$, onde p indica o valor do coeficiente igual ao polynomial_coefficient_K_lsc_R, polynomial_coefficient_K_lsc_G ou polynomial_coefficient_K_lsc_B, e r indica o valor do radio após a normalização por full_radius. N é igual ao valor de num_polynomial_coefficients_lsc .

[0072] num_deadzones é um número inteiro que especifica o número de zonas mortas na imagem codificada de cada amostra à qual esta caixa se aplica.

[0073] deadzone_left_horizontal_offset, deadzone_top_vertical_offset, deadzone_width, and deadzone_height são números inteiros que especificam a posição e tamanho da área da zona morta retangular na qual os pixels não são usados. deadzone_left_horizontal_offset and deadzone_top_vertical_offset especificam as coordenadas horizontal e vertical, respectivamente, em amostras luma, do canto esquerdo superior da zona morta na imagem codificada. deadzone_width e deadzone_height especificam a largura e altura, respectivamente, em amostras luma, da zona morta. Para salvar bits para representar o vídeo, todos os pixels em uma zona morta devem ser configurados com o mesmo valor de pixel, por exemplo, todos em preto.

[0074] As FIGs. 3-8 são diagramas conceituais que ilustram vários aspectos da sintaxe e semântica acima. A FIG. 3 ilustra a sintaxe da image_center_x e image_center_y como centro 302, a sintaxe full_radius como radio completo 304, picture_radius como raio do quadro 306 do quadro 300, scene_radius como raio da cena 308 (ex., a área sem obstruções do corpo da câmera 510). A FIG. 4

ilustra o displyed_fov (isto é, campo de visão exibido (FOV)) para duas imagens fisheye. FOV 402 representa um campo de visão de 170 graus e o FOV 404 representa um campo de visão de 190 graus. A FIG. 5 ilustra o displyed_fov (isto é, campo de visão exibido (FOV)) e overlapped_fov para múltiplas imagens fisheye (ex., N > 2). FOV 502 representa um primeiro campo de visão e o FOV 504 representa um segundo campo de visão. A FIG. 6 é uma ilustração da sintaxe camera_center_offset_x (o_{x*}), camera_center_offset_y (o_y), e camera_center_offset_z (o_z). A FIG. 7 é um diagrama conceitual ilustrando parâmetros com relação ao campo de visão local. A FIG. 8 é um diagrama conceitual ilustrando um exemplo de um campo de visão local.

[0075] A sinalização do vídeo fisheye no rascunho atual do OMAF DIS pode apresentar uma ou mais desvantagens.

[0076] Como um exemplo de desvantagem da sinalização do vídeo fisheye no rascunho atual do OMAF DIS, quando há duas imagens circulares em cada imagem de vídeo fisheye (por exemplo, onde num_circular_images na estrutura de sintaxe FisheyeOmnidirectionalVideoInfo( ) são iguais a 2), dependendo dos valores dos parâmetros extrínsecos da câmera (por exemplo, camera_center_yaw, camera_center_pitch, camera_center_roll, camera_center_offset_x, camera_center_offset_y e camera_center_offset_z), o vídeo fisheye pode ser monoscópico ou estereoscópico. Em particular, quando as duas câmeras que capturam as duas imagens circulares estão do mesmo lado, o vídeo fisheye é estereoscópico cobrindo

cerca da metade (ou seja, 180 graus horizontalmente) da esfera, e quando as duas câmeras estão em lados opostos, o vídeo fisheye é monoscópico, mas cobre aproximadamente a esfera inteira (ou seja, 360 graus horizontalmente). Por exemplo, quando dois conjuntos de valores de parâmetro extrínseco de câmera são os seguintes, o vídeo fisheye é monoscópico:

1° conjunto:

camera_center_yaw = 0 graus (+/- 5 graus)

camera_center_pitch = 0 graus (+/- 5 graus)

camera_center_roll = 0 graus (+/- 5 graus)

camera_center_offset_x = 0 mm (+/- 3 mm)

camera_center_offset_y = 0 mm (+/- 3 mm)

camera_center_offset_z = 0 mm (+/- 3 mm)

2° conjunto:

camera_center_yaw = 180 graus (+/- 5 graus)

camera_center_pitch = 0 graus (+/- 5 graus)

camera_center_roll = 0 graus (+/- 5 graus)

camera_center_offset_x = 0 mm (+/- 3 mm)

camera_center_offset_y = 0 mm (+/- 3 mm)

camera_center_offset_z = 0 mm (+/- 3 mm)

[0077] Os valores do parâmetro acima podem corresponder aos valores de parâmetro extrínseco da câmera do dispositivo de computação 10B da FIG. 1B. Como outro exemplo, quando dois conjuntos de valores de parâmetro extrínseco de câmera são os seguintes, o vídeo fisheye é estereoscópico:

1° conjunto:

camera_center_yaw = 0 graus (+/- 5 graus)

camera_center_pitch = 0 graus (+/- 5 graus)

camera_center_roll = 0 graus (+/- 5 graus)

camera_center_offset_x = 0 mm (+/- 3 mm)

camera_center_offset_y = 0 mm (+/- 3 mm)

camera_center_offset_z = 0 mm (+/- 3 mm)

2° conjunto:

camera_center_yaw = 0 graus (+/- 5 graus)

camera_center_pitch = 0 graus (+/- 5 graus)

camera_center_roll = 0 graus (+/- 5 graus)

camera_center_offset_x = 64 mm (+/- 3 mm)

camera_center_offset_y = 0 mm (+/- 3 mm)

camera_center_offset_z = 0 mm (+/- 3 mm)

[0078] Os valores do parâmetro acima podem corresponder aos valores de parâmetro extrínseco da câmera do dispositivo de computação 10A da FIG. 1 A. Observe que a distância X do deslocamento estéreo de 64 mm é equivalente a 2,5 polegadas, que é a distância média entre os olhos humanos.

[0079] Em outras palavras, as informações sobre se o vídeo fisheye é monoscópico ou estereoscópico estão ocultas (isto é, codificadas implicitamente, mas não explicitamente). No entanto, para fins de sistema de alto nível, como seleção de conteúdo, pode ser desejável que essas informações sejam facilmente acessíveis no nível do formato de arquivo e no DASH (por exemplo, para que a entidade que executa a função de seleção de conteúdo não precise analisar muito informações na estrutura da sintaxe FisheyeOmnidirectionalVideoInfo( ) para determinar se os dados de vídeo correspondentes são monoscópicos ou estereoscópicos).

[0080] Como outro exemplo de desvantagem da

sinalização do vídeo fisheye no rascunho atual do OMAF DIS, quando há duas imagens circulares em cada imagem de vídeo fisheye (por exemplo, onde imagens circulares na estrutura da sintaxe FisheyeOmnidirectionalVideoInfo( ) são iguais a 2) e o vídeo fisheye é estereoscópico, um dispositivo cliente pode determinar qual das duas imagens circulares é a visão do olho esquerdo e qual é a visão do olho direito dos parâmetros extrínsecos da câmera. No entanto, pode não ser desejável que o dispositivo cliente precise determinar qual imagem é a visão do olho esquerdo e qual é a visão do olho direito dos parâmetros extrínsecos da câmera.

[0081] Como outro exemplo de desvantagem da sinalização do vídeo fisheye no rascunho atual do OMAF DIS, quando há quatro câmeras fisheye, duas de cada lado, para capturar o vídeo fisheye, o vídeo fisheye seria estereoscópico cobrindo toda a esfera. Nesse caso, há quatro imagens circulares em cada imagem de vídeo fisheye (por exemplo, um número de imagens circulares na estrutura de sintaxe FisheyeOmnidirectionalVideoInfo( ) é igual a 4). Nesses casos (quando houver mais de duas imagens circulares em cada imagem de vídeo fisheye), um dispositivo cliente pode determinar o pareamento de certas duas imagens circulares pertencentes à mesma vista a partir dos parâmetros extrínsecos da câmera. No entanto, pode não ser desejável que o dispositivo cliente tenha que determinar o pareamento de certas duas imagens circulares pertencentes à mesma vista dos parâmetros extrínsecos da câmera.

[0082] Como outro exemplo de desvantagem da sinalização do vídeo fisheye no rascunho atual do OMAF DIS, para o vídeo omnidirecional projetado, as informações de

cobertura (por exemplo, qual área da esfera é coberta pelo vídeo) são explicitamente sinalizadas usando a CoverageInformationBox ou, quando não estiver presente, inferido pelo dispositivo cliente como sendo a esfera completa. Embora o dispositivo cliente possa determinar a cobertura a partir dos parâmetros extrínsecos da câmera, novamente, pode ser desejável a sinalização explícita dessas informações para serem facilmente acessíveis.

[0083] Como outro exemplo de desvantagem da sinalização de vídeo fisheye no rascunho atual do OMAF DIS, o uso de empacotamento por região é permitido para vídeo omnidirecional projetado, mas não permitido para vídeo fisheye. No entanto, alguns benefícios do empacotamento por região aplicável ao vídeo omnidirecional projetado também podem ser aplicáveis ao vídeo fisheye.

[0084] Como outro exemplo de desvantagem da sinalização de vídeo fisheye no rascunho atual do OMAF DIS, o transporte de vídeo omnidirecional projetado em faixas de subimagem é especificado no OMAF DIS usando o agrupamento de faixas de composição. No entanto, o transporte de vídeo omnidirecional fisheye em faixas de subimagem não é suportado.

[0085] Esta revelação apresenta soluções para os problemas acima. Algumas dessas técnicas podem ser aplicadas de forma independente e algumas delas podem ser aplicadas em combinação.

[0086] De acordo com uma ou mais técnicas desta revelação, um arquivo que inclui dados de vídeo fisheye pode incluir uma indicação explícita de se os dados de vídeo fisheye são monoscópicos ou estereoscópicos. Em

outras palavras, uma indicação de se o vídeo fisheye é monoscópico ou estereoscópico pode ser explicitamente sinalizada. Dessa maneira, um dispositivo cliente pode evitar ter que inferir se os dados de vídeo fisheye são monoscópicos ou estereoscópicos a partir de outros parâmetros.

[0087] Como exemplo, um ou mais dos 24 bits iniciais na estrutura da sintaxe FisheyeOmnidirectionalVideoInfo( ) podem ser usados para formar um campo (por exemplo, um sinalizador de um bit) para sinalizar a indicação de monoscópico ou estereoscópico. O campo sendo igual a um primeiro valor específico (por exemplo, 0) pode indicar que o vídeo fisheye é monoscópico e o campo igual a um segundo valor específico (por exemplo, 1) pode indicar que o vídeo fisheye é estereoscópico. Por exemplo, ao gerar um arquivo incluindo dados de vídeo fisheye, um dispositivo de preparação de conteúdo (por exemplo, dispositivo de preparação de conteúdo 20 da FIG. 9 e, em um exemplo específico, a unidade de encapsulamento 30 do dispositivo de preparação de conteúdo 20) pode codificar uma caixa (por exemplo, uma FisheyeOmnidirectionalVideoBox) dentro do arquivo, a caixa incluindo uma estrutura de sintaxe (por exemplo, uma FisheyeOmnidirectionalVideoInfo( )) que contém parâmetros para os dados de vídeo fisheye e a estrutura da sintaxe, incluindo uma indicação explícita de se os dados de vídeo fisheye são monoscópicos ou estereoscópicos. Um dispositivo cliente (ex., dispositivo cliente 40 na FIG. 9, e em um exemplo específico, a unidade de recuperação 52 do dispositivo cliente 40) pode processar da mesma forma o

arquivo para obter a indicação explícita de monoscópico ou estereoscópico.

[0088] Como outro exemplo, uma nova caixa contendo um campo, por exemplo, um sinalizador de um bit, alguns bits reservados e possivelmente alguma outra informação pode ser adicionada à FisheyeOmnidirectionalVideoBox para sinalizar a indicação de monoscópico ou estereoscópico. O campo sendo igual a um primeiro valor específico (por exemplo, 0) pode indicar que o vídeo fisheye é monoscópico e o campo igual a um segundo valor específico (por exemplo, 1) pode indicar que o vídeo fisheye é estereoscópico. Por exemplo, ao gerar um arquivo incluindo dados de vídeo fisheye, um dispositivo de preparação de conteúdo (por exemplo, dispositivo de preparação de conteúdo 20 da FIG. 9) pode codificar uma primeira caixa (por exemplo, uma FisheyeOmnidirectionalVideoBox) dentro do arquivo, a primeira caixa incluindo uma segunda caixa que inclui um campo que indica explicitamente se os dados de vídeo fisheye são monoscópicos ou estereoscópicos. Um dispositivo cliente (ex., dispositivo cliente 40 na FIG. 9) pode igualmente processar o arquivo para obter a indicação explícita de monoscópico ou estereoscópico.

[0089] Como outro exemplo, um campo (por exemplo, um sinalizador de um bit) pode ser adicionado diretamente à FisheyeOmnidirectionalVideoBox para sinalizar essa indicação. Por exemplo, ao gerar um arquivo incluindo dados de vídeo fisheye, um dispositivo de preparação de conteúdo (por exemplo, dispositivo de preparação de conteúdo 20 da FIG. 9) pode codificar uma caixa (por

exemplo, uma FisheyeOmnidirectionalVideoBox) dentro do arquivo, a caixa incluindo um campo que indica explicitamente se os dados de vídeo fisheye são monoscópicos ou estereoscópicos. Um dispositivo cliente (ex., dispositivo cliente 40 na FIG. 9) pode igualmente processar o arquivo para obter a indicação explícita de monoscópico ou estereoscópico.

[0090] De acordo com uma ou mais técnicas desta revelação, um arquivo que inclui dados de vídeo fisheye como uma pluralidade de imagens circulares pode incluir, para cada respectiva imagem circular da pluralidade de imagens circulares, uma indicação explícita de uma respectiva identificação de vista (ID da vista). Em outras palavras, para cada imagem circular de dados de vídeo fisheye, pode ser adicionado um campo que indica o ID da vista de cada imagem circular. Quando existem apenas dois valores de ID de vista 0 e 1, então a vista com ID de vista igual a 0 pode ser a vista esquerda e a vista com ID de vista igual a 1 pode ser a vista direita. Por exemplo, onde a pluralidade de imagens circulares inclui apenas duas imagens circulares, uma imagem circular da pluralidade de imagens circulares com uma primeira identificação de vídeo predeterminada é uma vista esquerda e uma imagem circular da pluralidade de imagens circulares com uma segunda identificação de vídeo predeterminada é uma vista direita. Dessa maneira, um dispositivo cliente pode evitar ter que inferir qual imagem circular é a vista esquerda e qual é a vista direita de outros parâmetros.

[0091] O ID da vista sinalizado também pode ser usado para indicar a relação de pareamento de quaisquer

duas imagens circulares pertencentes à mesma vista (por exemplo, como ambas podem ter o mesmo valor do ID da vista sinalizada).

[0092] Para permitir que os IDs de vista sejam acessados facilmente, sem a necessidade de analisar todos os parâmetros de vídeo fisheye para as imagens circulares, um loop pode ser adicionado após o campo num_circular_images e antes do loop existente. Por exemplo, o processamento do arquivo pode incluir a análise, em um primeiro loop, das indicações explícitas das identificações da vista; e análise, em um segundo loop posterior ao primeiro loop, de outros parâmetros das imagens circulares. Por exemplo, os IDs da vista e os outros parâmetros podem ser analisados da seguinte forma:

```
aligned(8) class FisheyeOmnidirectionalVideoInfo( ) {
    bit(24) reserved = 0;
    unsigned int(8) num_circular_images;
    for(i=0; i< num_circular_images; i++) {
        unsigned int(8) view_id;
        bit(24) reserved = 0;
    }
    for(i=0; i< num_circular_images; i++) {
        unsigned int(32) image_center_x;
        unsigned int(32) image_center_y;
    ...
    }
}
```

[0093] view_id indica o identificador da vista à qual a imagem circular pertence. Quando existem apenas dois valores de view_id 0 e 1 para todas as imagens circulares, as imagens circulares com view_id igual a 0 podem pertencer à vista esquerda e as imagens circulares com view_id igual a 1 podem pertencer à vista direita. Em alguns exemplos, o(s) elemento(s) da sintaxe que indica o identificador da vista (ou seja, o elemento da sintaxe que indica qual imagem circular é a vista esquerda e qual é a vista direita) pode ser o mesmo que o elemento da sintaxe que indica explicitamente se o dados de vídeo fisheye são estereoscópicos ou monoscópicos.

[0094] De acordo com uma ou mais técnicas desta revelação, um arquivo que inclui dados de vídeo fisheye pode incluir uma primeira caixa que inclui uma estrutura de sintaxe que contém parâmetros para os dados de vídeo fisheye e, opcionalmente, pode incluir uma segunda caixa que indica informações de cobertura para os dados de vídeo fisheye. Por exemplo, a sinalização das informações de cobertura para o vídeo omnidirecional fisheye pode ser adicionada à FisheyeOmnidirecionalVideoBox. Por exemplo, a CoverageInformationBox, conforme definida no OMAF DIS, pode ser opcionalmente contida na FisheyeOmnidirectionalVideoBox, e quando não estiver presente na FisheyeOmnidirectionalVideoBox, pode ser inferida uma cobertura total da esfera para o vídeo fisheye. Quando CoverageInformationBox está presente na FisheyeOmnidirectionalVideoBox, a região esférica representada pelas imagens de vídeo fisheye pode ser uma região especificada por dois círculos de guinada e dois

círculos de inclinação. Dessa maneira, um dispositivo cliente pode evitar ter que inferir a informação de cobertura a partir de outros parâmetros.

[0095] De acordo com uma ou mais técnicas desta revelação, um arquivo que inclui dados de vídeo fisheye pode incluir uma primeira caixa que inclui informações do esquema, a primeira caixa pode incluir uma segunda caixa que inclui uma estrutura de sintaxe que contém parâmetros para os dados de vídeo fisheye e a primeira caixa pode opcionalmente incluir uma terceira caixa que indica se as imagens dos dados de vídeo fisheye são empacotadas por região. Por exemplo, a RegionWisePackingBox pode ser opcionalmente incluída na SchemeInformationBox para vídeo omnidirecional fisheye (por exemplo, quando FisheyeOmnidirectionalVideoBox está presente na SchemeInformationBox, RegionWisePackingBox pode estar presente na mesma SchemeInformationBox). Nesse caso, a presença da RegionWisePackingBox indica que as imagens de vídeo fisheye são compactadas na região e exigem descompactação antes da renderização. Dessa maneira, um ou mais benefícios da empacotamente por região podem ser obtidos no vídeo fisheye.

[0096] De acordo com uma ou mais técnicas desta revelação, a composição de subimagem pode ser usada para agrupar vídeo omnidirecional fisheye, e uma especificação sobre a aplicação do agrupamento de composição de subimagem pode ser adicionada ao vídeo omnidirecional fisheye. Essa especificação pode ser aplicada quando qualquer uma das faixas mapeadas para o grupo de faixas de composição de subimagem tiver um tipo de

entrada de amostra igual a 'resv' e um tipo de esquema igual a 'fodv' na SchemeTypeBox incluído na entrada de amostra. Nesse caso, cada imagem composta é uma imagem empacotada que tem o formato fisheye indicado por qualquer FisheyeOmnidirectionalVideoBox e o formato de empacotamento por região indicado por qualquer RegionWisePackingBox incluída nas entradas de amostra das trilhas mapeadas para o grupo de faixa da composição de subimagem. Além disso, o seguinte pode ser aplicado:

[0097] 1) Cada faixa mapeada para este agrupamento deve ter um tipo de entrada de amostra igual a 'resv'. O tipo de esquema deve ser igual a 'fodv' no SchemeTypeBox incluído na entrada da amostra.

[0098] 2) O conteúdo de todas as instâncias da FisheyeOmnidirectionalVideoBox incluídas nas entradas de amostra das faixas mapeadas para o mesmo grupo de faixas de composição de subimagem deve ser idêntico.

[0099] 3) O conteúdo de todas as instâncias da RegionWisePackingBox incluídas nas entradas de amostra das faixas mapeadas para o mesmo grupo de faixas de composição de subimagem deve ser idêntico.

[00100] No streaming HTTP, operações mais frequentemente usadas incluem HEAD, GET, e GET parcial. A operação HEAD recupera um cabeçalho de um arquivo associado a um determinado localizador de recurso uniforme (URL) ou nome de recursos uniforme (URN), sem recuperar uma carga útil associada com o URL ou URN. A operação GET recupera um arquivo inteiro associado a um determinado URL ou URN. A operação GET parcial recebe uma faixa de bytes como um parâmetro de entrada e recupera um número contínuo de bytes

de um arquivo, em que o número de bytes corresponde à faixa de byte recebida. Assim, fragmentos de filmes podem ser fornecidos para fluxo contínuo de HTTP, porque uma operação GET parcial pode obter um ou mais fragmentos de filmes individuais. Em um fragmento de filme, pode haver vários fragmentos de faixa de diferentes faixas. No fluxo contínuo de HTTP, uma apresentação de mídia pode ser uma coleta estruturada de dados que é acessível ao cliente. O cliente pode solicitar e baixar informações de dados de mídia para apresentar um serviço de fluxo contínuo a um usuário.

[00101] No exemplo de fluxo contínuo de dados 3GPP que usa fluxo contínuo de HTTP, pode haver múltiplas representações para dados de vídeo e/ou de áudio do conteúdo multimídia. Como explicado abaixo, diferentes representações podem corresponder a diferentes características de codificação (por exemplo, diferentes perfis ou níveis de um padrão de codificação de vídeo), diferentes padrões ou extensões de codificação dos padrões de codificação (como extensões de multivista e/ou escaláveis), ou diferentes taxas de bits. O manifesto de tais representações pode ser definido em uma estrutura de dados de Descrição de Apresentação de Mídia (MPD). Uma apresentação de mídia pode corresponder a uma coleta estruturada de dados que é acessível a um dispositivo do cliente de fluxo contínuo de HTTP. O dispositivo do cliente de fluxo contínuo de HTTP pode solicitar e baixar informações de dados de mídia para apresentar um serviço de fluxo contínuo a um usuário do dispositivo do cliente. Uma apresentação de mídia pode ser descrita na estrutura de dados MPD, que pode incluir atualizações de MPD.

[00102] Uma apresentação em mídia pode conter uma sequência de um ou mais períodos. Cada período pode ir até ao início do próximo Período, ou até ao fim da apresentação de mídia, no caso do último período. Cada período pode conter uma ou mais representações para o mesmo conteúdo de mídia. Uma representação pode ser uma de uma série de versões codificadas alternativas de dados de áudio, vídeo, texto temporizado ou outros desses dados.. As representações podem diferir por tipos de codificação, por exemplo, pela taxa de bits, resolução e/ou codec de dados de vídeo e taxa de bit, idioma e/ou codec para dados de áudio. O termo representação pode ser utilizado para se referir a uma seção de dados de áudio ou de vídeo codificados correspondentes a um período particular do conteúdo de multimídia codificado de uma maneira específica.

[00103] Representações de um período específico podem ser atribuídas a um grupo indicado por um atributo no MPD indicativo de um conjunto de adaptação ao qual as representações pertencem. Representações no mesmo conjunto de adaptação são geralmente consideradas como substitutas uma para a outra, em que um dispositivo cliente pode dinamicamente e facilmente alternar entre essas representações, por exemplo, para realizar a adaptação da largura de banda. Por exemplo, cada representação dos dados de vídeo para um determinado período pode ser atribuída ao mesmo conjunto de adaptação, de modo que qualquer uma das representações pode ser selecionada para decodificação para apresentar dados multimídia, como dados de vídeo ou dados de áudio, do conteúdo multimídia para o período

correspondente. O conteúdo de mídia dentro de um período pode ser representado por qualquer representação do grupo 0, se presente, ou a combinação de, no máximo, uma representação de cada grupo diferente de zero, em alguns exemplos. Dados de temporização para cada representação de um período podem ser expressos em relação ao tempo de início do período.

[00104] Uma representação pode incluir um ou mais segmentos. Cada representação pode incluir um segmento de inicialização, ou cada segmento de uma representação pode ser de auto-inicialização. Quando presente, o segmento de inicialização pode conter informação de inicialização para acessa a representação. Em geral, o segmento de inicialização não contém dados de mídia. Um segmento pode ser referido unicamente por um identificador, como um localizador de recurso uniforme (URL), nome do recurso uniforme (URN), ou identificador de recurso uniforme (URI). A MPD pode fornecer os identificadores para cada segmento. Em alguns exemplos, a MPD também pode fornecer faixas de bytes na forma de um atributo de faixa, que pode corresponder aos dados para um segmento dentro de um arquivo acessível pelo URL, URN, ou URI.

[00105] Diferentes representações podem ser selecionadas para a recuperação substancialmente simultânea de diferentes tipos de dados de mídia. Por exemplo, um dispositivo cliente pode selecionar uma representação de áudio, uma representação de vídeo, e uma representação de texto cronometrada a partir da qual recuperar segmentos. Em alguns exemplos, o dispositivo cliente pode selecionar conjuntos específicos de adaptação para realizar a

adaptação da largura de banda. Ou seja, o dispositivo cliente pode selecionar um conjunto de adaptação incluindo representações de vídeo, um conjunto de adaptação incluindo representações de áudio e/ou um conjunto de adaptação incluindo texto cronometrado. Alternativamente, o dispositivo cliente pode selecionar conjuntos de adaptação para certos tipos de mídia (por exemplo, vídeo), e selecionar diretamente representações para outros tipos de mídia (por exemplo, áudio e/ou texto cronometrado).

[00106] FIG. 9 é um diagrama de blocos que ilustra um sistema exemplar 10 que implementa técnicas para fluxo contínuo de dados de mídia sobre uma rede. Neste exemplo, o sistema 10 inclui o dispositivo de preparação de conteúdo 20, o dispositivo servidor 60, e o dispositivo cliente 40. O dispositivo cliente 40 e dispositivo servidor 60 estão acoplados de forma comunicativa pela rede 74, que pode compreender a Internet. Em alguns exemplos, o dispositivo de preparação de conteúdo 20 e dispositivo servidor 60 também podem ser acoplados pela rede 74 ou outra rede, ou podem ser diretamente acoplados comunicativamente. Em alguns exemplos, o dispositivo de preparação de conteúdo 20 e dispositivo servidor 60 podem compreender o mesmo dispositivo.

[00107] O dispositivo de preparação de conteúdo 20, no exemplo da FIG. 9, compreende fonte de áudio 22 e fonte de vídeo 24. A fonte de áudio 22 pode compreender, por exemplo, um microfone que produz sinais elétricos representativos de dados de áudio capturados a serem codificados pelo codificador de áudio 26. Alternativamente, a fonte de áudio 22 pode compreender uma mídia de

armazenamento que armazena os dados de áudio previamente registados, um gerador de dados de áudio, como um sintetizador computadorizado, ou qualquer outra fonte de dados de áudio. A fonte de vídeo 24 pode compreender uma câmera de vídeo que produz dados de vídeo a serem codificados por um codificador de vídeo 28, uma mídia de armazenamento codificada com dados de vídeo previamente gravados, uma unidade de geração de dados de vídeo, como uma fonte de gráficos de computador, ou qualquer outra fonte de dados de vídeo. O dispositivo de preparação de conteúdo 20 não é necessariamente acoplado comunicativamente ao dispositivo servidor 60 em todos os exemplos, mas pode armazenar o conteúdo multimídia para uma mídia separada que é lida pelo dispositivo servidor 60.

[00108] Os dados de áudio e dados de vídeo podem compreender dados analógicos ou digitais. Os dados analógicos podem ser digitalizados antes de serem codificados pelo codificador de áudio 26 e/ou codificador de vídeo 28. A fonte de áudio 22 pode obter dados de áudio de um participante que está falando enquanto o participante que está falando está falando, e a fonte de vídeo 24 pode simultaneamente obter dados de vídeo do participante que está falando. Em outros exemplos, a fonte de áudio 22 pode compreender uma mídia de armazenamento legível por computador que compreende os dados de áudio armazenados, e a fonte de vídeo 24 pode compreender uma mídia de armazenamento de leitura por computador que compreende dados de vídeo armazenados. Deste modo, as técnicas descritas na presente revelação podem ser aplicadas aos dados de áudio e vídeo ao vivo, de fluxo contínuo e em

tempo real ou aos dados de áudio e vídeo arquivados, pré-gravados.

[00109] Os quadros de áudio que correspondem aos quadros de vídeo são geralmente quadros de áudio que contém dados de áudio que foram capturados (ou gerados) pela fonte de áudio 22 simultaneamente com dados de vídeo capturados (ou gerados) pela fonte de vídeo 24 que está contida dentro dos quadros de vídeo. Por exemplo, enquanto um participante que está falando geralmente produz dados de áudio pela fala, a fonte de áudio 22 capta os dados de áudio e a fonte de vídeo 24 captura os dados de vídeo do participante que está falando ao mesmo tempo, que é, enquanto a fonte de áudio 22 está capturando os dados de áudio. Assim, um quadro de áudio pode temporalmente corresponder a um ou mais quadros de vídeo específicos. Deste modo, um quadro de áudio correspondente a um quadro de vídeo geralmente corresponde a uma situação em que os dados de áudio e dados de vídeo foram capturados ao mesmo tempo e para a qual um quadro de áudio e um quadro de vídeo compreendem, respectivamente, os dados de áudio e os dados de vídeo que foram capturados ao mesmo tempo.

[00110] Em alguns exemplos, o codificador de áudio 26 pode codificar uma marca de hora em cada quadro de áudio codificado que representa um momento em que os dados de áudio para o quadro de áudio codificado foram registrados, e de forma semelhante, o codificador de vídeo 28 pode codificar uma marca de hora em cada quadro de vídeo codificado que representa um momento no qual os dados de vídeo do quadro de vídeo codificado foram gravados. Nesses exemplos, um quadro de áudio correspondente a um quadro de

vídeo pode compreender um quadro de áudio que compreende uma marca de hora e um quadro de vídeo que compreende a mesma marca de hora. O dispositivo de preparação de conteúdo 20 pode incluir um relógio interno a partir do qual o codificador de áudio 26 e/ou codificador de vídeo 28 pode gerar as marcas de hora, ou que a fonte de áudio 22 e fonte de vídeo 24 pode usar para associar dados de áudio e vídeo, respectivamente, com uma marca de hora.

[00111] Em alguns exemplos, a fonte de áudio 22 pode enviar dados para o codificador de áudio 26 correspondentes a um momento em que os dados de áudio foram gravados, e a fonte de vídeo 24 pode enviar dados para o codificador de vídeo 28 correspondente a um momento em que os dados de vídeo foram gravados. Em alguns exemplos, o codificador de áudio 26 pode codificar um identificador de sequência nos dados de áudio codificados para indicar uma ordenação temporal relativa dos dados de áudio codificados mas sem indicar necessariamente um tempo absoluto no qual os dados de áudio foram gravados, e de forma semelhante, um codificador de vídeo 28 também pode utilizar identificadores de sequências para indicar uma ordenação temporal relativa dos dados de vídeo codificados. Do mesmo modo, em alguns exemplos, um identificador de sequência pode ser mapeado ou de outra forma correlacionado com uma marca de tempo.

[00112] O codificador de áudio 26 geralmente produz um fluxo de dados de áudio codificados, enquanto o codificador de vídeo 28 produz um fluxo de dados de vídeo codificados. Cada fluxo individual de dados (seja de áudio ou vídeo) pode ser referido como um fluxo elementar. Um

fluxo elementar é um único componente codificado digitalmente (possivelmente comprimido) de uma representação. Por exemplo, a parte de vídeo ou de áudio codificada da representação pode ser um fluxo elementar. Um fluxo elementar pode ser convertido em um fluxo elementar em pacotes (PES), antes de ser encapsulado dentro de um arquivo de vídeo. Dentro da mesma representação, o ID do fluxo pode ser utilizado para distinguir os pacotes PES pertencentes a um fluxo elementar a partir do outro. A unidade básica de dados de um fluxo elementar é um pacote de fluxo elementar empacotado (PES). Assim, os dados de vídeo codificados geralmente correspondem aos fluxos de vídeo elementares. Da mesma forma, os dados de áudio correspondem a um ou mais respectivos fluxos elementares.

[00113] Muitos padrões de codificação de vídeo, como ITU-T H.264/AVC e o padrão de Codificação de Video de Alta Eficiência (HEVC), definem a sintaxe, semântica e processo de decodificação dos fluxos de bits livres de erros, qualquer um dos quais se adequa a um determinado perfil ou nível. Os padrões de codificação de vídeo geralmente não especificam o codificador, mas o codificador tem a tarefa de garantir que os fluxos de bits gerados são compatível com o padrão de um decodificador. No contexto dos padrões de codificação de vídeo, um "perfil" corresponde a um subconjunto de algoritmos, recursos ou ferramentas e restrições que lhes são aplicáveis. Como definido pelo padrão H.264, por exemplo, um "perfil" é um subconjunto de toda a sintaxe do fluxo de bits que é especificada pelo padrão H.264. Um "nível" corresponde às limitações do consumo de recursos do decodificador, como,

por exemplo, memória do decodificador e computação, que estão relacionados com a resolução das imagens, taxa de bits e taxa de processamento de bloco. Um perfil pode ser sinalizado com um valor de idc (indicador de perfil) do perfil, enquanto um nível pode ser sinalizado com um valor de idc (indicador de nível) do nível.

[00114] O padrão H.264, por exemplo, reconhece que, dentro dos limites impostos pela sintaxe de um determinado perfil, ainda é possível requerer uma grande variação no desempenho dos codificadores e decodificadores dependendo dos valores assumidos pelos elementos de sintaxe no fluxo de bits como o tamanho especificado das imagem decodificadas. O padrão H.264 reconhece ainda que, em muitas aplicações, não é nem prático nem econômico implementar um decodificador capaz de lidar com todos os usos hipotéticos da sintaxe dentro de um perfil particular. Assim, o padrão H.264 define um "nível" como um conjunto específico de restrições impostas aos valores dos elementos de sintaxe no fluxo de bit. Estas restrições podem ser simples limites em valores. Alternativamente, estas restrições podem assumir a forma de restrições sobre combinações aritméticas de valores (por exemplo, largura da imagem multiplicada pela altura da imagem multiplicado pelo número de imagens decodificadas por segundo). O padrão H.264 prevê ainda que implementações individuais podem suportar um nível diferente para cada perfil suportado.

[00115] Um decodificador de acordo com um perfil normalmente suporta todas as características definidas no perfil. Por exemplo, como uma característica de codificação, a codificação da imagem B não é suportada

no perfil da linha base do H.264/AVC, mas é suportada em outros perfis de H.264/AVC. Um decodificador de acordo com um nível deve ser capaz de decodificar qualquer fluxo de bits que não requer recursos além das limitações definidas no nível. As definições dos perfis e níveis pode ser útil para facilidade de interpretação. Por exemplo, durante a transmissão de vídeo, um par de definições de perfil e nível pode ser negociado e acordado para uma sessão de transmissão inteira. Mais especificamente, em H.264/AVC, um nível pode definir limitações sobre o número de macroblocos que precisam ser processados, tamanho do buffer da imagem decodificada (DPB), buffer da imagem codificada (CEC), faixa de vetor de movimento vertical, número máximo de vetores de movimento por dois MBs consecutivos, e se um bloco B pode ter partições de sub-macrobloco menores que 8x8 pixels. Desta maneira, um decodificador pode determinar se o decodificador é capaz de decodificar corretamente o fluxo de bits.

[00116] No exemplo da FIG. 9, a unidade de encapsulamento 30 do dispositivo de preparação de conteúdo 20 recebe fluxos elementares que compreendem dados de vídeo codificados do codificador de vídeo 28 e fluxos elementares que compreendem dados de áudio codificados do codificador de áudio 26. Em alguns exemplos, o codificador de vídeo 28 e o codificador de áudio 26 podem cada incluir empacotadores para formar pacotes PES a partir de dados codificados. Em outros exemplos, o codificador de vídeo 28 e o codificador de áudio 26 podem cada fazer interface com os respectivos empacotadores para formar pacotes PES a partir de dados codificados. Ainda em outros exemplos, a

unidade de encapsulamento 30 pode incluir empacotadores para formar pacotes PES de dados de áudio e vídeo codificados.

[00117] O codificador de vídeo 28 pode codificar dados de vídeo de conteúdo multimídia de várias maneiras, para produzir diferentes representações do conteúdo multimídia em várias taxas de bits e com várias características, como resoluções de pixel, taxas de quadros, conformidade aos vários padrões de codificação, conformidade aos vários perfis e/ou níveis de perfil para vários padrões de codificação, representações tendo um ou vários modos de exibição (por exemplo, para reprodução bidimensional ou tridimensional), ou outras dessas características. Uma representação, como utilizada na presente revelação, pode compreender um dos dados de áudio, dados de vídeo, dados de texto (por exemplo, para as legendas fechadas), ou outros dados semelhantes. A representação pode incluir um fluxo elementar, como um fluxo elementar de áudio ou um fluxo elementar de vídeo. Cada pacote PES pode incluir uma ID de fluxo que identifica o fluxo elementar ao qual pertence o pacote PES. A unidade de encapsulamento 30 é responsável pela montagem de fluxos elementares em arquivos de vídeo (por exemplo, segmentos) de diferentes representações.

[00118] A unidade de encapsulamento 30 recebe pacotes PES para fluxos elementares de uma representação do codificador de áudio 26 e codificador de vídeo 28 e forma unidades de camada de abstração de rede correspondentes (NAL) a partir dos pacotes PES. Segmentos de vídeo codificados podem ser organizados em unidades NAL, que

fornecem uma representação de vídeo "amigável à rede" abordando aplicações como telefonia de vídeo, armazenamento, broadcast ou fluxo contínuo. As unidades NAL podem ser categorizadas para unidades NAL de Camada de Codificação de Vídeo (VCL) e unidades NAL não-VCL. As unidades de VCL podem conter o motor de compressão de núcleo e podem incluir bloco, macrobloco, e/ou dados do nível de fatia. Outras unidades NAL podem ser unidades NAL não-VCL. Em alguns exemplos, uma imagem codificada em um instante de tempo, normalmente apresentada como uma imagem primária codificada, pode estar contida em uma unidade de acesso, que pode incluir uma ou mais unidades NAL.

[00119] As unidades NAL não-VCL podem incluir unidades NAL de conjunto de parâmetro e unidades NAL SEI, entre outras. Os conjuntos de parâmetros podem conter informações de cabeçalho de nível sequência (em conjuntos de parâmetros de sequência (SPS)) e as informações de cabeçalho de nível de imagem raramente em modificação (nos conjuntos de parâmetros de imagem (PPS)). Com conjuntos de parâmetros (por exemplo, PPS e SPS), as informações que raramente mudam não precisam ser repetidas para cada sequência ou imagem, portanto, a eficiência da codificação pode ser melhorada. Além disso, o uso de conjuntos de parâmetros pode permitir a transmissão fora da banda da informação de cabeçalho importante, evitando a necessidade de transmissões redundantes para a resiliência de erro. Nos exemplos de transmissão fora d banda, as unidades NAL do conjunto de parâmetro podem ser transmitidas em um canal diferente do que outras unidades NAL, como unidades NAL SEI.

[00120] As Informações de Melhoramento Suplementares (SEI) podem conter informações que não são necessárias para decodificar as amostras de imagens codificadas das unidades NAL VCL, mas podem ajudar em processos relacionados à decodificação, exibição, resiliência de erro, e outros fins. As mensagens SEI podem estar contidas em unidades NAL não-VCL. As mensagens SEI são a parte normativa de algumas especificações padrão e, portanto, nem sempre são obrigatórias para aplicação do decodificador compatível ao padrão. As mensagens SEI podem ser mensagens SEI de nível de sequência ou mensagens SEI de nível de imagem. Algumas informações sobre o nível de sequência podem estar contidas em mensagens SEI, como mensagens SEI de informações de escalabilidade no exemplo de SVC e mensagens de informação de escalabilidade de vista na MVC. Estas mensagens SEI exemplares podem transmitir informações sobre, por exemplo, a extração de pontos de funcionamento e características dos pontos de operação. Além disso, a unidade de encapsulamento 30 pode formar um arquivo de manifesto, como um descritor de apresentação de mídia (MPD) que descreve as características das representações. A unidade de encapsulamento 30 pode formatar o MPD de acordo com a linguagem de marcação extensível (XML).

[00121] A unidade de encapsulamento 30 pode fornecer dados para uma ou mais representações de conteúdo multimídia, juntamente com o arquivo de manifesto (por exemplo, o MPD) para interface de saída 32. A interface de saída 32 pode compreender uma interface de rede ou uma interface para gravar em uma mídia de armazenamento, como

uma interface de barramento em série universal (USB), um gravador de CD ou DVD, uma interface para mídias de armazenamento magnéticas ou flash, ou outras interfaces para armazenar ou transmitir dados de mídia. A unidade de encapsulamento 30 pode fornecer dados de cada uma das representações de conteúdo multimídia para a interface de saída 32, o que pode enviar os dados para o dispositivo servidor 60 através da transmissão de rede ou mídia de armazenamento. No exemplo da FIG. 9, o dispositivo servidor 60 inclui mídia de armazenamento 62 que armazena vários conteúdos multimídia 64, cada uma incluindo um respectivo arquivo de manifesto 66 e uma ou mais representações 68A-68N (representações 68). Em alguns exemplos, a interface de saída 32 também pode enviar dados diretamente para a rede 74.

[00122] Em alguns exemplos, as representações 68 podem ser separadas em conjuntos de adaptação. Ou seja, vários subconjuntos de representações 68 podem incluir respectivos conjuntos comuns de características, como codec, perfil e nível, resolução, número de vistas, formato de arquivo para segmentos, informações de tipo de texto que podem identificar uma linguagem ou outras características do texto a serem apresentadas com a representação e/ou dados de áudio a serem decodificados e apresentados, por exemplo, por alto falantes, informações de ângulo de câmera que podem descrever um ângulo da câmera ou perspectiva da câmera do mundo real de um cenário para representações no conjunto de adaptação, informações de classificação que descrevem a adequação do conteúdo para públicos específicos, ou semelhantes.

[00123] O arquivo de manifesto 66 pode incluir dados indicativos dos subconjuntos de representações 68 correspondentes a determinados conjuntos de adaptação, bem como características comuns para os conjuntos de adaptação. O arquivo de manifesto 66 também pode incluir dados representativos das características individuais, como taxas de bits, para as representações individuais de conjuntos de adaptação. Desta forma, um conjunto de adaptação pode fornecer uma adaptação da largura de banda da rede simplificada. Representações em um conjunto de adaptação podem ser indicadas através de elementos filho de um elemento do conjunto de adaptação do arquivo de manifesto 66. Em alguns exemplos, o arquivo de manifesto 66 pode incluir alguns ou todos dentre os dados de FisheyeOmnidirectionalVideoInfo( ) discutidos aqui, ou dados similares. Adicional ou alternativamente, os segmentos de representações 68 podem incluir alguns ou todos dentre os dados de FisheyeOmnidirectionalVideoInfo( ) discutidos aqui, ou dados similares.

[00124] O dispositivo de servidor 60 inclui a unidade de processamento de pedido 70 e interface de rede 72. Em alguns exemplos, o dispositivo servidor 60 pode incluir uma pluralidade de interfaces de rede. Além disso, todo e qualquer recurso do dispositivo servidor 60 pode ser implementado em outros dispositivos de uma rede de entrega de conteúdo, como roteadores, pontes, dispositivos de proxy, comutadores, ou outros dispositivos. Em alguns exemplos, os dispositivos intermediários de uma rede de entrega de conteúdo podem armazenar em cache dados de conteúdos multimídia 64, e incluir componentes que

substancialmente se adequam com aqueles do dispositivo servidor 60. Em geral, a interface de rede 72 é configurada para enviar e receber dados através da rede 74.

[00125] A unidade de processamento de pedido 70 é configurada para receber solicitações de rede de dispositivos clientes, como dispositivo cliente 40, para os dados da mídia de armazenamento 62. Por exemplo, a unidade de processamento de pedido 70 pode implementar o protocolo de transferência de hipertexto (HTTP) versão 1.1, conforme descrito em RFC 2616, "Hypertext Transfer Protocol - HTTP/1.1," por R. Fielding et al, Grupo de Trabalho de Rede, IETF, Junho de 1999. Ou seja, a unidade de processamento de pedido 70 pode ser configurada para receber pedidos GET de HTTP ou GET parciais e fornecer dados de conteúdos multimídia 64, em resposta aos pedidos. Os pedidos podem especificar um segmento de uma das representações 68, por exemplo, utilizando uma URL do segmento. Em alguns exemplos, os pedidos também podem especificar uma ou mais faixas de bytes do segmento, compreendendo assim solicitações GET parciais. A unidade de processamento de pedido 70 pode ainda ser configurada para pedidos HEAD de HTTP de serviço para fornecer dados de cabeçalho de um segmento de uma das representações 68. Em qualquer caso, a unidade de processamento do pedido 70 pode ser configurada para processar os pedidos para fornecer dados solicitados para um dispositivo solicitante, como dispositivo cliente 40.

[00126] Adicional ou alternativamente, a unidade de processamento de pedido 70 pode ser configurada para entregar dados de mídia através de um protocolo de

broadcast ou multicast, como eMBMS. Dispositivo de preparação de conteúdo 20 pode criar segmentos e/ou sub-segmentos DASH substancialmente da mesma maneira como descrito, mas o dispositivo servidor 60 pode fornecer estes segmentos ou sub-segmentos usando eMBMS ou outro protocolo de transporte de rede de broadcast ou multicast. Por exemplo, a unidade de processamento de pedido 70 pode ser configurada para receber um pedido de participação do grupo multicast a partir do dispositivo cliente 40. Ou seja, o dispositivo servidor 60 pode anunciar um endereço de protocolo da Internet (IP) associado a um grupo multicast para dispositivos clientes, incluindo o dispositivo cliente 40, associado com conteúdo de mídia específico (por exemplo, uma transmissão de um evento ao vivo). O dispositivo cliente 40, por sua vez, pode apresentar um pedido para se juntar ao grupo multicast. Esse pedido pode ser propagado em toda a rede 74, por exemplo, os roteadores que compõem a rede 74, de modo que faz-se com que os roteadores direcionem o tráfego destinado para o endereço IP associado ao grupo de multicast aos dispositivos clientes assinantes, como o dispositivo cliente 40.

[00127] Como ilustrado no exemplo da FIG. 9, o conteúdo multimídia 64 inclui arquivo de manifesto 66, que pode corresponder a uma descrição de apresentação de mídia (MPD). O arquivo de manifesto 66 pode conter descrições de diferentes representações alternativas 68 (por exemplo, serviços de vídeo com diferentes qualidades) e a descrição pode incluir, por exemplo, informações de codec, um valor de perfil, um valor de nível, uma taxa de bits, e outras características descritivas das representações 68. O

dispositivo cliente 40 pode recuperar o MPD de uma apresentação de mídia para determinar como acessar segmentos de representações 68.

[00128] Em particular, a unidade de recuperação 52 pode recuperar os dados de configuração (não representados) do dispositivo cliente 40 para determinar as capacidades de decodificação do decodificador de vídeo 48 e capacidades de renderização da saída de vídeo 44. Os dados de configuração também podem incluir qualquer um ou todos de uma preferência de linguagem selecionada por um usuário do dispositivo cliente 40, uma ou mais perspectivas de câmara correspondentes às preferências de profundidade definidas pelo usuário do dispositivo cliente 40, e/ou uma preferência de classificação selecionada pelo usuário do dispositivo cliente 40. A unidade de recuperação de 52 pode compreender, por exemplo, um navegador da Web ou um cliente de mídia configurado para enviar solicitações GET e de GET parcial do HTTP. A unidade de recuperação 52 pode corresponder às instruções de software executadas por um ou mais processadores ou unidades de processamento (não mostradas) do dispositivo cliente 40. Em alguns exemplos, todas ou partes da funcionalidade descrita com relação à unidade de recuperação 52 podem ser implementadas em hardware, ou uma combinação de hardware, software e/ou firmware, onde o hardware do requisito pode ser fornecido para executar instruções para o software ou firmware.

[00129] A unidade de recuperação 52 pode comparar as capacidades de decodificação e de renderização do dispositivo cliente 40 às características de representações 68 indicadas pelas informações do arquivo de

manifesto 66. Por exemplo, a unidade de recuperação 52 pode determinar se o dispositivo cliente 40 (como a saída de vídeo) é capaz de renderizar os dados estereoscópicos ou somente dados monoscópicos. A unidade de recuperação 52 pode inicialmente recuperar pelo menos uma porção do arquivo de manifesto 66 para determinar as características de representações 68. Por exemplo, a unidade de recuperação 52 pode solicitar uma porção do arquivo de manifesto 66 que descreve características de um ou mais conjuntos de adaptação. A unidade de recuperação 52 pode selecionar um subconjunto de representações 68 (por exemplo, um conjunto de adaptação) que tem características que podem ser satisfeitas pelas capacidades de codificação e renderização do dispositivo cliente 40. A unidade de recuperação 52 pode, então, determinar taxas de bits para representações no conjunto de adaptação, determinar um montante atualmente disponível da largura de banda de rede e retirar segmentos de uma das representações que têm uma taxa de bits que pode ser satisfeita pela largura de banda da rede. De acordo com as técnicas desta revelação, a unidade de recuperação 52 pode recuperar dados indicando se os dados de vídeo fisheye correspondentes são monoscópicos ou estereoscópicos. Por exemplo, a unidade de recuperação 52 pode recuperar uma parte inicial de um arquivo (por exemplo, um segmento) de uma das representações 68 para determinar se o arquivo inclui dados de vídeo fisheye monoscópicos ou estereoscópicos e determinar se deseja recuperar dados de vídeo do arquivo, ou um arquivo diferente de uma representação diferente 68, de acordo com se o dispositivo cliente 40 é ou não capaz de renderizar os dados de vídeo

fisheye do arquivo.

[00130] Em geral, representações de taxa de bits mais elevadas podem produzir a reprodução de vídeo de qualidade superior, enquanto representações de taxa de bit inferior podem oferecer uma reprodução de vídeo com qualidade suficiente quando a largura de banda disponível da rede diminui. Assim, quando a largura de banda de rede disponível é relativamente alta, a unidade de recuperação 52 pode recuperar dados de representações de taxas de bits relativamente altas, enquanto quando a largura de banda disponível da rede é baixa, a unidade de recuperação 52 pode recuperar dados de representações taxas de bits relativamente baixas. Desta forma, o dispositivo cliente 40 pode transmitir dados multimídia através da rede 74, enquanto também se adapta às mudanças da disponibilidade de largura de banda de rede 74.

[00131] Adicional ou alternativamente, a unidade de recuperação 52 pode ser configurada para receber dados de acordo com um protocolo de rede de broadcast ou multicast, como eMBMS ou multicast IP. Nesses exemplos, a unidade de recuperação 52 pode apresentar um pedido para se juntar a um grupo de rede multicast associado com conteúdo de mídia específico. Depois de entrar para o grupo multicast, a unidade de recuperação 52 pode receber dados do grupo multicast sem pedidos adicionais emitidos para o dispositivo servidor 60 ou dispositivo de preparação de conteúdos 20. A unidade de recuperação 52 pode apresentar um pedido para deixar o grupo multicast quando os dados do grupo multicast não forem mais necessários, por exemplo, para interromper a reprodução ou para mudar canais para um

grupo multicast diferente.

[00132] A interface de rede 54 pode receber e fornecer dados de segmentos de uma representação selecionada para a unidade de recuperação 52, que por sua vez pode fornece os segmentos para a unidade de desencapsulação 50. A unidade de desencapsulação 50 pode desencapsular elementos de um arquivo de vídeo em fluxos PES constituintes, desempacotar os fluxos PES para recuperar dados codificados, e enviar os dados codificados ou para o decodificador de áudio 46 ou decodificador de vídeo 48, dependendo se os dados codificados são parte de uma sequência de áudio ou vídeo, por exemplo, como indicado por cabeçalhos dos pacotes PES do fluxo. O decodificador de áudio 46 decodifica os dados de áudio codificados e envia os dados de áudio decodificados para a saída de áudio 42, enquanto o decodificador de vídeo 48 decodifica os dados de vídeo codificados e envia os dados de vídeo decodificados, que podem incluir uma pluralidade de vistas de um fluxo, para a saída de vídeo 44.

[00133] O codificador de vídeo 28, decodificador de vídeo 48, codificador de áudio 26, decodificador de áudio 46, unidade de encapsulamento 30, unidade de recuperação 52 e a unidade de desencapsulação 50 cada um pode ser implementado como qualquer um de uma variedade de circuitos de processamento adequados, conforme o caso, como um ou mais microprocessadores, processadores sinal digital (DSPs), circuitos integrados de aplicação específica (ASICs), arranjos de portas programáveis em campo (FPGA), circuitos lógicos discretos, software, hardware, firmware ou quaisquer suas combinações. Cada

codificador de vídeo 28 e decodificador de vídeo 48 pode ser incluído em um ou mais codificadores ou decodificadores, cada um dos quais pode ser integrado como parte de um codificador/decodificador de vídeo combinados (codec). Do mesmo modo, cada um do codificador de vídeo 26 e decodificador de vídeo 46 pode ser incluído em um ou mais codificadores ou decodificadores, cada um dos quais pode ser integrado como parte de um CODEC combinado. Um equipamento, incluindo um codificador de vídeo 28, decodificador de vídeo 48, codificador de áudio 26, decodificador de áudio 46, unidade de encapsulação 30, unidade de recuperação 52 e/ou unidade de desencapsulamento 50 pode compreender um circuito integrado, um microprocessador e/ou um dispositivo de comunicação sem fio, como um telefone celular.

[00134] O dispositivo cliente 40, dispositivo servidor 60, e/ou dispositivo de preparação de conteúdo 20 podem ser configurados para operar de acordo com as técnicas desta revelação. Para fins de exemplo, esta revelação descreve estas técnicas no que diz respeito ao dispositivo cliente 40 e dispositivo servidor 60. No entanto, deve ser compreendido que o dispositivo de preparação de conteúdo 20 pode ser configurado para executar estas técnicas, em vez de (ou além do) dispositivo servidor 60.

[00135] A unidade de encapsulação 30 pode formar unidades NAL compreendendo um cabeçalho que identifica um programa ao qual a unidade NAL pertence, bem como uma carga útil, por exemplo, dados de áudio, dados de vídeo, ou dados que descreve o fluxo de transporte ou de

programa ao qual a unidade NAL corresponde. Por exemplo, em H.264/AVC, uma unidade NAL inclui um cabeçalho de 1-byte e uma carga útil de tamanho variável. Uma unidade NAL incluindo dados de vídeo em sua carga útil pode compreender vários níveis de granularidade de dados de vídeo. Por exemplo, uma unidade NAL pode compreender um bloco de dados de vídeo, uma pluralidade de blocos, uma fatia de dados de vídeo, ou uma imagem inteira de dados de vídeo. A unidade de encapsulação 30 pode receber dados vídeo codificados do codificador de vídeo 28, na forma de pacotes PES de fluxos elementares. A unidade de encapsulamento 30 pode associar cada fluxo elementar com um programa correspondente.

[00136] A unidade de encapsulamento 30 também pode montar unidades de acesso a partir de uma pluralidade de unidades NAL. Em geral, uma unidade de acesso pode compreender uma ou mais unidades NAL para representar um quadro de dados de vídeo, como também dados de áudio correspondentes ao quadro quando tais dados de áudio estão disponíveis. Uma unidade de acesso geralmente inclui todas as unidades NAL para uma instante de tempo de saída, por exemplo, todos os dados de áudio e vídeo para uma instante de tempo. Por exemplo, se cada vista tem uma taxa de quadro de 20 quadros por segundo (fps), em seguida, cada instante de tempo pode corresponder a um intervalo de tempo de 0,05 segundos. Durante este intervalo de tempo, os quadros específicos para todas as vistas da mesma unidade de acesso (o mesmo instante de tempo) podem ser processados simultaneamente. Em um exemplo, uma unidade de acesso pode compreender uma imagem codificada em um instante de tempo, que pode ser apresentado como uma imagem codificada

primária.

[00137] Consequentemente, uma unidade de acesso pode compreender todos os quadros de áudio e de vídeo de um instante temporal comum, por exemplo, todas as vistas correspondentes ao momento X. Esta revelação também se refere a uma imagem codificada de uma vista específica como um "componente de visualização". Ou seja, um componente de visualização pode incluir uma imagem codificada (ou quadro) para uma vista específica em um momento específico. Assim, uma unidade de acesso pode ser definida como compreendendo todos os componentes de visualização de um instante temporal comum. A ordem de decodificação das unidades de acesso não precisa ser necessariamente a mesma que a de saída ou ordem de exibição.

[00138] Uma apresentação de mídia pode incluir uma descrição de apresentação de mídia (MPD), a qual pode conter descrições de diferentes representações alternativas (por exemplo, serviços de vídeo com diferentes qualidades) e a descrição pode incluir, por exemplo, informações de codec, um valor de perfil, e um valor do nível. Uma MPD é um exemplo de um arquivo de manifesto, como o arquivo de manifesto 66. O dispositivo cliente 40 pode recuperar o MPD de uma apresentação de mídia para determinar como acessar fragmentos de filme das várias apresentações. Os fragmentos de filme podem ser localizados nas caixas de fragmento de filme (moof boxes) dos arquivos de vídeo.

[00139] O arquivo de manifesto 66 (o qual pode compreender, por exemplo, uma MPD) pode anunciar a disponibilidade dos segmentos das representações 68. Ou seja, a MPD pode incluir informações que indicam o tempo do

relógio de parede no qual um primeiro segmento de uma das representações 68 se torna disponível, assim como a informação que indica as durações dos segmentos dentro das representações 68. Desta maneira, a unidade de recuperação 52 do dispositivo cliente 40 pode determinar quando cada segmento está disponível, com base no tempo de início assim como as durações dos segmentos que precedem um segmento específico.

[00140] Após a unidade de encapsulamento 30 ter montado unidades NAL e/ou unidades de acesso em um arquivo de vídeo com base nos dados recebidos, a unidade de encapsulamento 30 passa o arquivo de vídeo para interface de saída 32 para a saída. Em alguns exemplos, a unidade de encapsulação 30 pode armazenar o arquivo de vídeo localmente ou enviar o arquivo de vídeo para um servidor remoto através da interface de saída 32, em vez de enviar o arquivo de vídeo diretamente para o dispositivo cliente 40. A interface de saída 32 pode compreender, por exemplo, um transmissor, um transceptor, um dispositivo para gravação de dados em uma mídia legível por computador, como, por exemplo, uma unidade ótica, uma unidade de mídia magnética (por exemplo, unidade de disquete), um barramento em série universal (USB), uma interface de rede, ou outra interface de saída. A interface de saída 32 emite o arquivo de vídeo para uma mídia legível por computador, como, por exemplo, um sinal de transmissão, uma mídia magnética, uma mídia ótica, uma memória, uma unidade flash ou outra mídia legível por computador.

[00141] A interface de rede 54 pode receber uma unidade NAL ou unidade de acesso através da rede 74 e

fornecer a unidade NAL ou unidade de acesso para a unidade de desencapsulação 50, através da unidade de recuperação 52. A unidade de desencapsulação 50 pode desencapsular elementos de um arquivo de vídeo em fluxos PES constituintes, desempacotar os fluxos PES para recuperar dados codificados, e enviar os dados codificados ou para o decodificador de áudio 46 ou decodificador de vídeo 48, dependendo se os dados codificados são parte de uma sequência de áudio ou vídeo, por exemplo, como indicado por cabeçalhos dos pacotes PES do fluxo. O decodificador de áudio 46 decodifica os dados de áudio codificados e envia os dados de áudio decodificados para a saída de áudio 42, enquanto o decodificador de vídeo 48 decodifica os dados de vídeo codificados e envia os dados de vídeo decodificados, que podem incluir uma pluralidade de vistas de um fluxo, para a saída de vídeo 44.

[00142] Desta maneira, o dispositivo cliente 40 representa um exemplo de um dispositivo para recuperar dados de mídia, o dispositivo incluindo um dispositivo para recuperar arquivos de dados de vídeo fisheye, como descrito acima e/ou conforme reivindicado abaixo. Por exemplo, o dispositivo cliente 40 pode recuperar arquivos de dados de vídeo fisheye e/ou renderizar vídeo fisheye com base em uma determinação de se o vídeo fisheye é monoscópico ou estereoscópico e essa determinação pode ser baseada em um elemento de sintaxe que especifica explicitamente se o vídeo fisheye é monoscópico ou estereoscópico.

[00143] Da mesma forma, o dispositivo de preparação de conteúdo 20 representa um dispositivo para gerar arquivos de dados de vídeo fisheye, como descrito

acima e/ou conforme reivindicado abaixo. Por exemplo, o dispositivo de preparação de conteúdo 20 pode incluir, em dados de vídeo fisheye, um elemento de sintaxe que especifica explicitamente se o vídeo fisheye é monoscópico ou estereoscópico.

[00144] A FIG. 10 é um diagrama conceitual ilustrando elementos do conteúdo multimídia exemplar 120. O conteúdo multimídia 120 pode corresponder aos conteúdos multimídia 64 (FIG. 9), ou outro conteúdo multimídia armazenado na mídia de armazenamento 62. No exemplo da FIG. 10, o conteúdo multimídia 120 inclui a descrição de apresentação de mídia (MPD) 122 e uma pluralidade de representações 124A-124N (representações 124). A representação 124A inclui dados de cabeçalho opcionais 126 e segmentos 128A-128N (segmentos 128), enquanto a representação 124N inclui dados de cabeçalho opcionais 130 e segmentos 132A-132N (segmentos 132). A letra N é utilizada para designar o último fragmento de filme em cada uma das representações 124, por uma questão de conveniência. Em alguns exemplos, pode haver um número diferente de fragmentos de filme entre as representações 124.

[00145] A MPD 122 pode compreender uma estrutura de dados separada das representações 124. A MPD 122 pode corresponder ao arquivo de manifesto 66 da FIG. 9. Em geral, a MPD 122 pode incluir dados que geralmente descrevem características de representações 124, como características de codificação e renderização, conjuntos de adaptação, um perfil ao qual a MPD 122 corresponde, informações de tipo texto, informação de ângulo da câmera,

informações de classificação, informações de modo de truque (por exemplo, informações indicativas das representações que incluem subsequências temporais), e/ou informações para recuperar períodos remotos (por exemplo, para a inserção de propaganda direcionada para conteúdo de mídia durante a reprodução).

[00146] Os dados do cabeçalho 126, quando presentes, podem descrever as características de segmentos 128, por exemplo, locais temporais de pontos de acesso aleatórios (RAPs, também conhecidos como pontos de acesso do fluxo (SAPs)), quais dos segmentos 128 inclui pontos de acesso aleatório, deslocamentos de byte para os pontos de acesso aleatório dentro dos segmentos 128, localizadores de recursos uniformes (URL) dos segmentos 128, ou outros aspectos de segmentos 128. Os dados de cabeçalho 130, quando presentes, podem descrever características semelhantes para os segmentos 132. Adicionalmente ou alternativamente, essas características podem ser totalmente incluídas dentro da MPD 122.

[00147] Os segmentos 128, 132 incluem uma ou mais amostras de vídeo codificadas, cada uma das quais podem incluir quadros ou fatias de dados de vídeo. Cada uma das amostras de vídeo codificadas dos segmentos 128 pode ter características semelhantes, por exemplo, requisitos de altura, largura e de largura de banda. Essas características podem ser descritas pelos dados da MPD 122, embora tais dados não sejam ilustrados no exemplo da FIG. 10. A MPD 122 pode incluir as características como descrito pela especificação do 3GPP, com a adição de qualquer ou toda a informação sinalizada descrita na presente

revelação.

[00148] Cada um dos segmentos 128, 132 pode ser associado a um localizador de recursos uniforme único (URL). Assim, cada um dos segmentos 128, 132 pode ser, independentemente recuperável utilizando um protocolo de rede de fluxo contínuo, como DASH. Desta maneira, um dispositivo de destino, como o dispositivo cliente 40, pode utilizar um pedido GET HTTP para recuperar segmentos 128 ou 132. Em alguns exemplos, o dispositivo cliente 40 pode usar solicitações GET HTTP parciais para recuperar intervalos de bytes específicos dos segmentos 128 ou 132.

[00149] A FIG. 11 é um diagrama de blocos que ilustra elementos de um arquivo de vídeo de exemplo 150, que pode corresponder a um segmento de uma representação, como um dos segmentos 114, 124 da FIG. 10. Cada um dos segmentos 128, 132 pode incluir dados que se ajustam substancialmente à disposição dos dados ilustrados no exemplo da FIG. 11. Pode-se dizer que o arquivo de vídeo 150 encapsula um segmento. Como descrito acima, os arquivos de vídeo de acordo com o formato de arquivo de mídia base ISO e suas extensões armazenam dados em uma série de objetos, chamados de "caixas". No exemplo da FIG. 11, o arquivo de vídeo 150 inclui caixa de tipo arquivo (FTYP) 152, caixa de filme (MOOV) 154, caixas de índice de segmento (sidx) 162, caixas de fragmento de filme (MOOF) 164 e caixa de acesso aleatório de fragmentos de filme (MFRA) 166. Embora a FIG. 11 represente um exemplo de um arquivo de vídeo, deve ser entendido que outros arquivos de mídia podem incluir outros tipos de dados de mídia (por exemplo, dados de áudio, dados de texto programados ou

similares) que são estruturados de maneira semelhante aos dados do arquivo de vídeo 150, de acordo com o formato de arquivo de mídia base ISO e suas extensões.

[00150] A caixa tipo de arquivo (FTYP) 152 geralmente descreve um tipo de arquivo para o arquivo de vídeo 150. A caixa de tipo de arquivo 152 pode incluir dados que identificam uma especificação que descreve uma melhor utilização para o arquivo de vídeo 150. A caixa de tipo arquivo 152 pode, alternativamente, ser colocada antes da caixa MOOV 154, das caixas de fragmentos de filme 164 e/ou da caixa MFRA 166.

[00151] Em alguns exemplos, um segmento, como o arquivo de vídeo 150, pode incluir uma caixa de atualização MPD (não mostrada) antes da caixa FTYP 152. A caixa de atualização do MPD pode incluir informações indicando que um MPD correspondente a uma representação incluindo o arquivo de vídeo 150 deve ser atualizado, juntamente com informações para atualizar o MPD. Por exemplo, a caixa de atualização do MPD pode fornecer um URI ou URL para um recurso a ser usado para atualizar o MPD. Como outro exemplo, a caixa de atualização de MPD pode incluir dados para atualizar o MPD. Em alguns exemplos, a caixa de atualização de MPD pode seguir imediatamente uma caixa do tipo de segmento (STYP) (não mostrada) do arquivo de vídeo 150, onde a caixa STYP pode definir um tipo de segmento para o arquivo de vídeo 150.

[00152] A caixa MOOV 154, no exemplo 2 da FIG. 11, inclui a caixa do cabeçalho do filme (MVHD) 156, a caixa da faixa (TRAK) 158 e uma ou mais caixas de extensão de filme (MVEX) 160. Em geral, a caixa MVHD 156 pode

descrever características gerais do arquivo de vídeo 150. Por exemplo, a caixa MVHD 156 pode incluir dados que descrevem quando o arquivo de vídeo 150 foi criado originalmente, quando o arquivo de vídeo 150 foi modificado pela última vez, uma escala de tempo para o arquivo de vídeo 150, uma duração de reprodução do arquivo de vídeo 150 ou outros dados que geralmente descrevem o arquivo de vídeo 150.

[00153] A caixa TRAK 158 pode incluir dados para uma faixa do arquivo de vídeo 150. A caixa TRAK 158 pode incluir uma caixa de cabeçalho de faixa (TKHD) que descreve características da faixa correspondente à caixa TRAK 158. Em alguns exemplos, a caixa TRAK 158 pode incluir imagens de vídeo codificadas, enquanto em outros exemplos, as imagens de vídeo codificadas da faixa podem ser incluídas nos fragmentos de filme 164, que podem ser referenciados por dados da caixa TRAK 158 e/ou caixas sidx 162.

[00154] Em alguns exemplos, o arquivo de vídeo 150 pode incluir mais de uma faixa. Por conseguinte, a caixa MOOV 154 pode incluir um número de caixas TRAK iguais ao número de faixas no arquivo de vídeo 150. A caixa TRAK 158 pode descrever características de uma faixa do arquivo de vídeo correspondente 150. Por exemplo, a caixa TRAK 158 pode descrever informações temporais e/ou espaciais para a faixa correspondente. Uma caixa TRAK semelhante à caixa TRAK 158 da caixa MOOV 154 pode descrever características de uma faixa de conjunto de parâmetros, quando a unidade de encapsulamento 30 (FIG. 9) inclui uma faixa de conjunto de parâmetros em um arquivo de vídeo, como o arquivo de vídeo

150. A unidade de encapsulamento 30 pode sinalizar a presença de mensagens SEI de nível de sequência na faixa de conjunto de parâmetros dentro da caixa TRAK que descreve a faixa de conjunto de parâmetros.

[00155] As caixas MVEX 160 podem descrever características dos fragmentos de filme correspondentes 164, por exemplo, para sinalizar que o arquivo de vídeo 150 inclui fragmentos de filme 164, além dos dados de vídeo incluídos na caixa MOOV 154, se houver. No contexto de streaming de dados de vídeo, imagens de vídeo codificadas podem ser incluídas nos fragmentos de filme 164 em vez de na caixa MOOV 154. Consequentemente, todas as amostras de vídeo codificadas podem ser incluídas nos fragmentos de filme 164 em vez de na caixa MOOV 154.

[00156] A caixa MOOV 154 pode incluir um número de caixas MVEX 160 iguais ao número de fragmentos de filme 164 no arquivo de vídeo 150. Cada uma das caixas MVEX 160 pode descrever características de um dos fragmentos de vídeo correspondentes 164. Por exemplo, cada caixa MVEX pode incluir uma caixa de cabeçalho de extensão de filme (MEHD) que descreve uma duração temporal para um correspondente dos fragmentos de filme 164.

[00157] Além disso, de acordo com as técnicas desta revelação, o arquivo de vídeo 150 pode incluir uma FisheyeOmnidirectionalVideoBox em uma SchemeInformationBox, que pode ser incluída na caixa MOOV 154. Em alguns exemplos, a FisheyeOmnidirectionalVieoBox pode ser incluída na caixa TRAK 158, se diferentes faixas do arquivo de vídeo 150 podem incluir dados de vídeo fisheye monoscópicos ou estereoscópicos. Em alguns exemplos, a

FisheyeOmnidirectionalVieoBox pode ser incluído na caixa do FOV 157.

[00158] Como observado acima, a unidade de encapsulamento 30 pode armazenar um conjunto de dados de sequência em uma amostra de vídeo que não inclui dados de vídeo codificados reais. Uma amostra de vídeo geralmente pode corresponder a uma unidade de acesso, que é uma representação de uma imagem codificada em uma instância de tempo específica. No contexto do AVC, a imagem codificada pode incluir uma ou mais unidades VCL NAL que contêm as informações para construir todos os pixels da unidade de acesso e outras unidades NAL não-VCL associadas, como mensagens SEI. Por conseguinte, a unidade de encapsulamento 30 pode incluir um conjunto de dados de sequência, que pode incluir mensagens SEI no nível de sequência, em um dos fragmentos de filme 164. A unidade de encapsulamento 30 pode ainda sinalizar a presença de um conjunto de dados de sequência e/ou mensagens SEI no nível de sequência como estando presentes em um dos fragmentos de filme 164 dentro de uma das caixas MVEX 160 correspondentes à dos fragmentos de filme 164.

[00159] As caixas SIDX 162 são elementos opcionais do arquivo de vídeo 150. Ou seja, os arquivos de vídeo em conformidade com o formato de arquivo 3 GPP ou outros formatos de arquivo não incluem necessariamente as caixas SIDX 162. De acordo com o exemplo do formato de arquivo 3GPP, uma caixa SIDX pode ser usada para identificar um subsegmento de um segmento (por exemplo, um segmento contido no arquivo de vídeo 150). O formato de arquivo 3GPP define um subsegmento como "um conjunto

independente de uma ou mais caixas de fragmentos de filmes consecutivas com as caixas de dados de mídia correspondentes e uma caixa de dados de mídia contendo dados referenciados por uma caixa de fragmento de filme deve seguir o fragmento de filme e preceder a próxima caixa de fragmento de filme que contém informações sobre a mesma faixa". O formato de arquivo 3GPP também indica que uma caixa SIDX "contém uma sequência de referências a subsegmentos do (sub)segmento documentado pela caixa. Os subsegmentos referenciados são contíguos no tempo de apresentação. Da mesma forma, os bytes referidos por uma caixa de índice de segmento são sempre contíguos no segmento. O tamanho referenciado fornece a contagem do número de bytes no material referenciado".

[00160] As caixas SIDX 162 geralmente fornecem informações representativas de um ou mais subsegmentos de um segmento incluído no arquivo de vídeo 150. Por exemplo, essas informações podem incluir tempos de reprodução nos quais os subsegmentos começam e/ou terminam, desvios de bytes para os subsegmentos, se os subsegmentos incluem (por exemplo, começam com) um ponto de acesso ao fluxo (SAP), um tipo para o SAP (por exemplo, se o SAP é uma imagem de atualização instantânea do decodificador (IDR), uma imagem limpa de acesso aleatório (CRA), uma imagem de acesso a link quebrado (BLA) ou similar), uma posição da SAP (em termos de tempo de reprodução e/ou deslocamento de bytes) no subsegmento e similares.

[00161] Os fragmentos de filme 164 podem incluir uma ou mais imagens de vídeo codificadas. Em alguns exemplos, os fragmentos de filme 164 podem incluir um ou

mais grupos de imagens (GOPs), cada um dos quais pode incluir um número de imagens de vídeo codificadas, por exemplo, quadros ou imagens. Além disso, como descrito acima, os fragmentos de filme 164 podem incluir conjuntos de dados de sequência em alguns exemplos. Cada um dos fragmentos de filme 164 pode incluir uma caixa de cabeçalho de fragmento de filme (MFHD, não mostrada na FIG. 11). A caixa MFHD pode descrever características do fragmento de filme correspondente, como um número de sequência para o fragmento de filme. Os fragmentos de filme 164 podem ser incluídos na ordem do número de sequência no arquivo de vídeo 150.

[00162] A caixa MFRA 166 pode descrever pontos de acesso aleatórios dentro dos fragmentos de filme 164 do arquivo de vídeo 150. Isso pode ajudar na execução de modos de truque, como executar buscas em locais temporais específicos (isto é, tempos de reprodução) dentro de um segmento encapsulado pelo arquivo de vídeo 150. A caixa MFRA 166 é geralmente opcional e não precisa ser incluída em arquivos de vídeo, em alguns exemplos. Da mesma forma, um dispositivo cliente, como o dispositivo cliente 40, não precisa necessariamente fazer referência à caixa MFRA 166 para decodificar e exibir corretamente os dados de vídeo do arquivo de vídeo 150. A caixa MFRA 166 pode incluir um número de caixas de acesso aleatório ao fragmento de faixa (TFRA) (não mostrado) igual ao número de faixas do arquivo de vídeo 150 ou, em alguns exemplos, igual ao número de faixas de mídia (por exemplo, faixas sem sugestão)) do arquivo de vídeo 150.

[00163] Em alguns exemplos, os fragmentos de

filme 164 podem incluir um ou mais pontos de acesso ao fluxo (SAPs), como imagens IDR. Da mesma forma, a caixa MFRA 166 pode fornecer indicações de locais no arquivo de vídeo 150 dos SAPs. Por conseguinte, uma subsequência temporal do arquivo de vídeo 150 pode ser formada a partir de SAPs do arquivo de vídeo 150. A subsequência temporal também pode incluir outras figuras, como quadros P e/ou quadros B que dependem dos SAPs. Os quadros e/ou fatias da subsequência temporal podem ser dispostos dentro dos segmentos, de modo que os quadros/fatias da subsequência temporal que dependem de outros quadros/fatias da subsequência possam ser decodificados adequadamente. Por exemplo, no arranjo hierárquico de dados, os dados usados para previsão para outros dados também podem ser incluídos na subsequência temporal.

[00164] A FIG. 12 é um fluxograma que ilustra uma técnica de exemplo para processar um arquivo que inclui dados de vídeo fisheye, de acordo com uma ou mais técnicas desta revelação. As técnicas da FIG. 12 são descritas como sendo realizadas pelo dispositivo cliente 40 da FIG. 9, embora os dispositivos com configurações diferentes do dispositivo cliente 40 possam executar a técnica da FIG. 12.

[00165] O dispositivo cliente 40 pode receber um arquivo incluindo dados de vídeo fisheye e uma estrutura de sintaxe que inclui uma pluralidade de elementos de sintaxe que especificam atributos dos dados de vídeo fisheye (1202). Como discutido acima, a pluralidade de elementos de sintaxe pode incluir um primeiro elemento de sintaxe que indica explicitamente se os dados de vídeo

fisheye são monoscópicos ou estereoscópicos e um ou mais elementos de sintaxe que indicam implicitamente se os dados de vídeo fisheye são monoscópicos ou estereoscópicos. Os um ou mais elementos de sintaxe que indicam implicitamente se os dados de vídeo fisheye são monoscópicos ou estereoscópicos podem ser elementos de sintaxe que indicam explicitamente parâmetros extrínsecos das câmeras usadas para capturar os dados de vídeo fisheye. Em alguns exemplos, os um ou mais elementos de sintaxe podem ser, ou podem ser semelhantes a, elementos de sintaxe incluídos na estrutura de sintaxe FisheyeOmnidirectionalVideoInfo( ) no rascunho atual do OMAF DIS. Em alguns exemplos, o primeiro elemento da sintaxe pode ser incluído em um conjunto de bits iniciais da estrutura da sintaxe (por exemplo, 24 bits iniciais da estrutura da sintaxe FisheyeOmnidirectionalVideoInfo( )).

[00166] O dispositivo cliente 40 pode determinar, com base no primeiro elemento de sintaxe, se os dados de vídeo fisheye são monoscópicos ou estereoscópicos (1204). Um valor do primeiro elemento de sintaxe pode explicitamente indicar se os dados de vídeo fisheye são monoscópicos ou estereoscópicos. Como um exemplo, onde o primeiro elemento de sintaxe tem um primeiro valor, o dispositivo cliente 40 pode determinar que os dados de vídeo fisheye são monoscópicos (ou seja, que as imagens circulares incluídas nas imagens dos dados de vídeo fisheye têm eixos ópticos alinhados e estão voltados para direções opostas) . Como outro exemplo, onde o segundo elemento de sintaxe tem um primeiro valor, o dispositivo cliente 40 pode determinar que os dados de vídeo fisheye são

estereoscópicos (ou seja, que as imagens circulares incluídas nas imagens dos dados de vídeo fisheye têm eixos ópticos paralelos e estão voltados para a mesma direção) .

[00167] Como discutido acima, embora possa ser possível para o dispositivo cliente 40 determinar se os dados de vídeo fisheye são monoscópicos ou estereoscópicos com base nos elementos de sintaxe que indicam explicitamente parâmetros extrínsecos das câmeras usadas para capturar os dados de vídeo fisheye, tal cálculo pode aumentar a carga computacional no dispositivo cliente 40. Dessa forma, para reduzir os cálculos realizados (e, portanto, os recursos computacionais utilizados), o dispositivo cliente 40 pode determinar se os dados de vídeo fisheye são monoscópicos ou estereoscópicos com base no primeiro elemento de sintaxe.

[00168] O dispositivo cliente 40 pode renderizar os dados de vídeo fisheye com base na determinação. Por exemplo, em resposta à determinação de que os dados de vídeo fisheye são monoscópicos, o dispositivo cliente 40 pode renderizar os dados fisheye como monoscópicos (1206). Por exemplo, o dispositivo cliente 40 pode determinar uma janela de visão (isto é, uma porção da esfera na qual um usuário está "olhando"), identificar uma porção das imagens circulares dos dados de vídeo fisheye que correspondem à janela de visão e exibir o mesma porção das imagens circulares para cada um dos olhos do espectador. De modo similar, em resposta à determinação de que os dados de vídeo fisheye são estereoscópicos, o dispositivo cliente 40 pode renderizar os dados fisheye como estereoscópicos (1208). Por exemplo, o dispositivo

cliente 40 pode determinar uma janela de visão (isto é, uma porção da esfera na qual um usuário está "olhando"), identificar uma respectiva porção de cada uma das imagens circulares dos dados de vídeo fisheye que correspondem à janela de visão e exibir as respectivas porções das imagens circulares para os olhos do espectador.

[00169] A FIG. 13 é um fluxograma que ilustra uma técnica de exemplo para gerar um arquivo que inclui dados de vídeo fisheye, de acordo com uma ou mais técnicas desta revelação. As técnicas da FIG. 13 são descritas como sendo realizadas pelo dispositivo de preparação de conteúdo 20 da FIG. 9, embora os dispositivos com configurações diferentes do dispositivo de preparação de conteúdo 20 possam executar a técnica da FIG. 13.

[00170] O dispositivo de preparação de conteúdo 20 pode obter dados de vídeo fisheye e parâmetros extrínsecos de câmeras usadas para capturar os dados de vídeo fisheye (1302). Por exemplo, o dispositivo de preparação de conteúdo 20 pode obter imagens dos dados de vídeo fisheye que são codificadas usando um codec de vídeo, como HEVC. Cada uma das imagens dos dados de vídeo fisheye pode incluir uma pluralidade de imagens circulares que correspondem a uma imagem capturada por uma câmera diferente com uma lente fisheye (por exemplo, uma imagem pode incluir uma primeira imagem circular capturada através da lente fisheye 12A da FIG. 1A e uma segunda imagem circular capturada através da lente fisheye 12B da FIG. 1A). Os parâmetros extrínsecos podem especificar vários atributos das câmeras. Por exemplo, os parâmetros extrínsecos podem especificar um ângulo de guinada, um

ângulo de inclinação, um ângulo de rotação e um ou mais deslocamentos espaciais de cada uma das câmeras usadas para capturar os dados de vídeo fisheye.

[00171] O dispositivo de preparação de conteúdo 20 pode determinar, com base nos parâmetros extrínsecos, se os dados de vídeo fisheye são monoscópicos ou estereoscópicos (1304). Como um exemplo, quando dois conjuntos de valores de parâmetro extrínseco de câmera são os seguintes, o dispositivo de preparação de conteúdo 20 pode determinar que o vídeo fisheye é estereoscópico:

1° conjunto:

camera_center_yaw = 0 graus (+/- 5 graus)

camera_center _pitch = 0 graus (+/- 5 graus)

camera_center roll = 0 graus (+/- 5 graus)

camera_center_offset_x = 0 mm (+/- 3 mm)

camera_center_offset_y = 0 mm (+/- 3 mm)

camera_center_offset_z = 0 mm (+/- 3 mm)

2° conjunto:

camera_center_yaw = 0 graus (+/- 5 graus)

camera_center _pitch = 0 graus (+/- 5 graus)

camera_center roll = 0 graus (+/- 5 graus)

camera_center_offset_x = 64 mm (+/- 3 mm)

camera_center_offset_y = 0 mm (+/- 3 mm)

camera_center_offset_z = 0 mm (+/- 3 mm)

[00172] Como outro exemplo, quando dois conjuntos de valores de parâmetro extrínseco de câmera são os seguintes, o dispositivo de preparação de conteúdo 20 pode determinar que o vídeo fisheye é monoscópico:

1° conjunto:

camera_center_yaw = 0 graus (+/- 5 graus)

camera_center _pitch = 0 graus (+/- 5 graus)

camera_center roll = 0 graus (+/- 5 graus)

camera_center_offset_x = 0 mm (+/- 3 mm)

camera_center_offset_y = 0 mm (+/- 3 mm)

camera_center_offset_z = 0 mm (+/- 3 mm)

2° conjunto:

camera_center_yaw = 180 graus (+/- 5 graus)

camera_center _pitch = 0 graus (+/- 5 graus)

camera_center roll = 0 graus (+/- 5 graus)

camera_center_offset_x = 0 mm (+/- 3 mm)

camera_center_offset_y = 0 mm (+/- 3 mm)

camera_center_offset_z = 0 mm (+/- 3 mm)

[00173] O dispositivo de preparação de conteúdo 20 pode codificar em um arquivo, os dados de vídeo fisheye e uma estrutura de sintaxe incluindo uma pluralidade de elementos de sintaxe que especificam atributos dos dados de vídeo fisheye (1306). A pluralidade de elementos de sintaxe incluem: um primeiro elemento de sintaxe que indica explicitamente se os dados de vídeo fisheye são monoscópicos ou estereoscópicos, e um ou mais elementos de sintaxe que indicam explicitamente os parâmetros extrínsecos das câmeras usadas para capturar os dados de vídeo fisheye.

[00174] Em um ou mais exemplos, as funções descritas podem ser implementadas em hardware, software, firmware ou qualquer combinação dos mesmos. Se implementadas em software, as funções podem ser armazenadas em ou transmitidas através de uma ou mais instruções ou código em uma mídia legível por computador e executadas por uma unidade de processamento baseada em hardware. A mídia

legível por computador pode incluir a mídia de armazenamento legível por computador, que corresponde a uma mídia tangível, como mídia de armazenamento de dados ou mídia de comunicação, incluindo qualquer meio que facilite a transferência de um programa de computador de um lugar para outro, por exemplo, de acordo com um protocolo de comunicação. Deste modo, a mídia legível por computador pode geralmente corresponder a (1) mídia de armazenamento legível por computador tangível que é não-transitória ou (2) um meio de comunicação, como um sinal ou onda transportadora. A mídia de armazenamento de dados pode ser qualquer mídia disponível que pode ser acessada por um ou mais computadores ou um ou mais processadores para recuperar instruções, código, e/ou estruturas de dados para a implementação das técnicas descritas na presente revelação. Um produto de programa de computador pode incluir uma mídia legível por computador.

[00175] A título de exemplo, e não como limitação, tais mídias de armazenamento legíveis por computador podem compreender RAM, ROM, EEPROM, CD-ROM ou outro armazenamento em disco ótico, armazenamento em disco magnético ou outros dispositivos de armazenamento magnéticos, memória flash ou qualquer outro meio que possa ser utilizado para armazenar código de programa desejado sob a forma de instruções ou estruturas de dados e que pode ser acessado por um computador. Também, qualquer conexão é adequadamente chamada de uma mídia legível por computador. Por exemplo, se as instruções são transmitidas a partir de um site, servidor, ou de outra origem remota através de um cabo coaxial, cabo de fibra óptica, par trançado, linha de

assinante digital (DSL), ou tecnologias sem fios, tais como infravermelho, rádio e microondas, então o cabo coaxial, cabo de fibra óptica, par trançado, DSL, ou tecnologias sem fios, tais como infravermelho, rádio e microondas estão incluídas na definição de mídia de transmissão. Deve-se compreender, no entanto, que a mídia de armazenamento legível por computador e mídia de armazenamento de dados não incluem conexões, ondas transportadoras, sinais ou outra mídia transitória, mas são, ao invés, direcionadas à mídia de armazenamento tangível, não transitória. Disco e disquete, como aqui utilizados, incluem disco compacto (CD), disco a laser, disco ótico, disco versátil digital (DVD), disquete e disco Blu-ray onde os disquetes geralmente reproduzem dados magneticamente, enquanto que os discos reproduzem dados oticamente com lasers. Combinações dos anteriores também devem ser incluídas dentro do escopo de mídias legíveis por computador.

[00176] As instruções podem ser executadas por um ou mais processadores, como um ou mais processadores de sinal digital (DSPs), microprocessadores de uso geral, circuitos integrados de aplicação específica (ASIC), arranjos de lógica programáveis em campo (FPGA), ou outros circuitos lógicos ou discretos equivalentes. Consequentemente, o termo "processador" como aqui utilizado pode referir-se a qualquer uma das estruturas precedentes ou qualquer outra estrutura adequada para implementação das técnicas aqui descritas. Além disso, em alguns aspectos, a funcionalidade aqui descrita pode ser fornecida dentro de módulos de hardware e/ou software dedicados configurados para a codificação e decodificação, ou incorporados em um

codec combinado. Também, as técnicas podem ser totalmente implementadas em um ou mais circuitos ou elementos lógicos.

[00177] As técnicas da presente revelação podem ser implementadas em uma vasta variedade de dispositivos ou aparelhos, incluindo um monofone sem fios, um circuito integrado (IC) ou um conjunto de ICs (por exemplo, um conjunto de chips). Vários componentes, módulos ou unidades são descritos na presente revelação para enfatizar os aspectos funcionais dos dispositivos configurados para executar as técnicas divulgadas, mas não precisam necessariamente da realização por diferentes unidades de hardware. Em vez disso, como descrito acima, várias unidades podem ser combinadas em uma unidade de hardware de codec ou fornecida por uma coleta de unidades de hardware interoperativas, incluindo um ou mais processadores, conforme descrito acima, em conjunto com o software e/ou firmware adequado.

[00178] Vários exemplos foram descritos. Esses e outros exemplos estão dentro do escopo das reivindicações a seguir.

REIVINDICAÇÕES

1. Um método de processamento de um arquivo incluindo dados de vídeo, o método compreendendo: processar um arquivo incluindo dados de vídeo fisheye, o arquivo incluindo uma estrutura de sintaxe incluindo uma pluralidade de elementos de sintaxe que especificam atributos dos dados de vídeo fisheye, em que a pluralidade dos elementos de sintaxe inclui: um primeiro elemento de sintaxe que indica explicitamente se os dados de vídeo fisheye são monoscópicos ou estereoscópicos; e um ou mais elementos de sintaxe que implicitamente indicam se os dados de vídeo de fisheye são monoscópicos ou estereoscópicos.

determinar, com base no primeiro elemento de sintaxe, se os dados de vídeo de fisheye são monoscópicos ou estereoscópicos; e

entregar, com base na determinação, os dados de vídeo fisheye para renderização como monoscópicos ou estereoscópicos.

2. O método, de acordo com a reivindicação 1, em que o primeiro elemento de sintaxe é incluído em um conjunto de bits iniciais da estrutura de sintaxe.

3. O método, de acordo com a reivindicação 2, sendo que o conjunto de bits iniciais tem 24 bits de comprimento.

4. O método, de acordo com a reivindicação 1, em que o arquivo inclui uma caixa que inclui a estrutura de sintaxe.

5. O método, de acordo com a reivindicação 4, em que a caixa é uma primeira caixa que é incluída em uma segunda caixa que inclui as informações do esquema, o

método compreende adicionalmente:

determinar se a primeira caixa inclui uma terceira caixa que indica se as imagens dos dados de vídeo fisheye são empacotadas de região em região.

6. O método, de acordo com a reivindicação 5, compreendendo ainda:

em resposta à determinação de que a primeira caixa inclui a terceira caixa, descompactar as imagens dos dados de vídeo fisheye antes de renderizar as imagens dos dados de vídeo fisheye; ou

em resposta à determinação de que a primeira caixa não inclui a terceira caixa, renderizar as imagens dos dados de vídeo fisheye sem desempacotar as imagens dos dados de vídeo fisheye.

7. O método, de acordo com a reivindicação 5, em que a primeira caixa é uma SchemeInformationBox, a segunda caixa é uma FisheyeOmnidirectionalVideoBox e uma terceira caixa é uma Region WisePackingBox.

8. O método, de acordo com a reivindicação 1, em que a estrutura de sintaxe inclui ainda um segundo elemento de sintaxe que especifica um número de imagens circulares incluídas em cada imagem dos dados de vídeo fisheye.

9. O método, de acordo com a reivindicação 8, em que a estrutura de sintaxe compreende, para cada respectiva imagem circular, um respectivo terceiro elemento de sintaxe que indica um identificador de vista da respectiva imagem circular.

10. O método, de acordo com a reivindicação 1, em que a estrutura da sintaxe é externa aos dados da camada de codificação de vídeo (VCL) encapsulados pelo arquivo.

11. O método, de acordo com a reivindicação 1, sendo que determinar se os dados de vídeo fisheye são monoscópicos ou estereoscópicos compreende:

determinar, com base no primeiro elemento de sintaxe e independentemente dos elementos de sintaxe que implicitamente indicam se os dados de vídeo fisheye são monoscópicos ou estereoscópicos, se os dados de vídeo fisheye são monoscópicos ou estereoscópicos.

12. Um método para gerar um arquivo incluindo dados de vídeo, o método compreendendo: obter os dados de vídeo fisheye e parâmetros extrínsecos das câmeras usadas para capturar os dados de vídeo fisheye;

determinar, com base nos parâmetros extrínsecos, se os dados de vídeo de fisheye são monoscópicos ou estereoscópicos; e

codificar, em um arquivo, os dados de vídeo fisheye e uma estrutura de sintaxe incluindo uma pluralidade de elementos de sintaxe que especificam atributos dos dados de vídeo fisheye, em que a pluralidade dos elementos de sintaxe inclui: um primeiro elemento de sintaxe que indica explicitamente se os dados de vídeo fisheye são monoscópicos ou estereoscópicos; e um ou mais elementos de sintaxe que explicitamente indicam os parâmetros extrínsecos das câmeras usadas para capturar os dados de vídeo fisheye.

13. O método, de acordo com a reivindicação 12, em que codificar o elemento de sintaxe compreende codificar o primeiro elemento de sintaxe em um conjunto de bits iniciais da estrutura de sintaxe.

14. O método, de acordo com a reivindicação 13,

sendo que o conjunto de bits iniciais tem 24 bits de comprimento.

15. O método, de acordo com a reivindicação 12, em que o arquivo inclui uma caixa que inclui a estrutura de sintaxe.

16. O método, de acordo com a reivindicação 15, em que a caixa é uma primeira caixa que é incluída em uma segunda caixa que inclui as informações do esquema, o método compreende adicionalmente:

codificar, na primeira caixa, uma terceira caixa que indica se as imagens dos dados de vídeo fisheye são empacotadas de região em região.

17. O método, de acordo com a reivindicação 16, em que a primeira caixa é uma SchemeInformationBox, a segunda caixa é uma FisheyeOmnidirectionalVideoBox e uma terceira caixa é uma Region WisePackingBox.

18. O método, de acordo com a reivindicação 12, em que a estrutura de sintaxe inclui ainda um segundo elemento de sintaxe que especifica um número de imagens circulares incluídas em cada imagem dos dados de vídeo fisheye.

19. O método, de acordo com a reivindicação 18, em que a estrutura de sintaxe compreende, para cada respectiva imagem circular, um respectivo terceiro elemento de sintaxe que indica um identificador de vista da respectiva imagem circular.

20. O método, de acordo com a reivindicação 12, em que os dados de vídeo fisheye são codificados em uma camada de codificação de vídeo (VCL) e em que a estrutura de sintaxe é externa à VCL.

21. Um dispositivo para processamento de dados de vídeo, o dispositivo compreendendo:

uma memória configurada para armazenar pelo menos uma porção de um arquivo incluindo dados de vídeo fisheye, o arquivo incluindo uma estrutura de sintaxe incluindo uma pluralidade de elementos de sintaxe que especificam atributos dos dados de vídeo fisheye, em que a pluralidade de elementos de sintaxe inclui: um primeiro elemento de sintaxe que indica explicitamente se os dados de vídeo fisheye são monoscópicos ou estereoscópicos, e um ou mais elementos de sintaxe que implicitamente indicam se os dados de vídeo fisheye são monoscópicos ou estereoscópicos; e

um ou mais processadores configurados para:

determinar, com base no primeiro elemento de sintaxe, se os dados de vídeo fisheye são monoscópicos ou estereoscópicos; e

entregar, com base na determinação, os dados de vídeo fisheye para renderização como monoscópicos ou estereoscópicos.

22. O dispositivo, de acordo com a reivindicação 21, em que o primeiro elemento de sintaxe é incluído em um conjunto de bits iniciais da estrutura de sintaxe.

23. O dispositivo, de acordo com a reivindicação 22, sendo que o conjunto de bits iniciais tem 24 bits de comprimento.

24. O dispositivo, de acordo com a reivindicação 21, em que o arquivo inclui uma caixa que inclui a estrutura de sintaxe.

25. O dispositivo, de acordo com a reivindicação 24, em que a caixa é uma primeira caixa que é incluída em

uma segunda caixa que inclui as informações do esquema, os um ou mais processadores são ainda configurados para:

determinar se a primeira caixa inclui uma terceira caixa que indica se as imagens dos dados de vídeo fisheye são empacotadas de região em região.

26. O dispositivo, de acordo com a reivindicação 25, sendo que:

em resposta à determinação de que a primeira caixa inclui a terceira caixa, os um ou mais processadores são ainda configurados para, descompactar as imagens dos dados de vídeo fisheye antes de renderizar as imagens dos dados de vídeo fisheye; ou

em resposta à determinação de que a primeira caixa não inclui a terceira caixa, os um ou mais processadores são ainda configurados para, renderizar as imagens dos dados de vídeo fisheye sem desempacotar as imagens dos dados de vídeo fisheye.

27. O dispositivo, de acordo com a reivindicação 25, em que a primeira caixa é uma SchemeInformationBox, a segunda caixa é uma FisheyeOmnidirectionalVideoBox e uma terceira caixa é uma Region WisePackingBox.

28. O dispositivo, de acordo com a reivindicação 21, em que a estrutura de sintaxe inclui ainda um segundo elemento de sintaxe que especifica um número de imagens circulares incluídas em cada imagem dos dados de vídeo fisheye.

29. O dispositivo, de acordo com a reivindicação 21, em que a estrutura da sintaxe é externa aos dados da camada de codificação de vídeo (VCL) encapsulados pelo arquivo.

30. O dispositivo, de acordo com a reivindicação 21, em que, para determinar se os dados de vídeo fisheye são monoscópicos ou estereoscópicos, um ou mais processadores são configurados para:

determinar, com base no primeiro elemento de sintaxe e independentemente dos elementos de sintaxe que implicitamente indicam se os dados de vídeo fisheye são monoscópicos ou estereoscópicos, se os dados de vídeo fisheye são monoscópicos ou estereoscópicos.

31. Um dispositivo para gerar um arquivo incluindo dados de vídeo, o dispositivo compreendendo: uma memória configurada para armazenar dados de vídeo fisheye; e

um ou mais processadores configurados para:

obter parâmetros extrínsecos de câmeras usadas para capturar os dados de vídeo fisheye;

determinar, com base nos parâmetros extrínsecos, se os dados de vídeo de fisheye são monoscópicos ou estereoscópicos; e

codificar, em um arquivo, os dados de vídeo fisheye e uma estrutura de sintaxe incluindo uma pluralidade de elementos de sintaxe que especificam atributos dos dados de vídeo fisheye, em que a pluralidade dos elementos de sintaxe inclui: um primeiro elemento de sintaxe que indica explicitamente se os dados de vídeo fisheye são monoscópicos ou estereoscópicos; e um ou mais elementos de sintaxe que explicitamente indicam os parâmetros extrínsecos das câmeras usadas para capturar os dados de vídeo fisheye.

32. O dispositivo, de acordo com a reivindicação

31, em que para codificar o primeiro elemento de sintaxe, os um ou mais processadores são configurados para codificar o primeiro elemento de sintaxe em um conjunto de bits iniciais da estrutura de sintaxe.

33. O dispositivo, de acordo com a reivindicação 32, sendo que o conjunto de bits iniciais tem 24 bits de comprimento.

34. O dispositivo, de acordo com a reivindicação 31, em que o arquivo inclui uma caixa que inclui a estrutura de sintaxe.

35. O dispositivo, de acordo com a reivindicação 34, em que a caixa é uma primeira caixa que é incluída em uma segunda caixa que inclui as informações do esquema, os um ou mais processadores são ainda configurados para:

codificar, na primeira caixa, uma terceira caixa que indica se as imagens dos dados de vídeo fisheye são empacotadas de região em região.

36. O dispositivo, de acordo com a reivindicação 35, em que a primeira caixa é uma SchemeInformationBox, a segunda caixa é uma FisheyeOmnidirectionalVideoBox e uma terceira caixa é uma Region WisePackingBox.

37. O dispositivo, de acordo com a reivindicação 31, em que a estrutura de sintaxe inclui ainda um segundo elemento de sintaxe que especifica um número de imagens circulares incluídas em cada imagem dos dados de vídeo fisheye.

38. O dispositivo, de acordo com a reivindicação 31, em que os dados de vídeo fisheye são codificados em uma camada de codificação de vídeo (VCL) e em que a estrutura de sintaxe é externa à VCL.

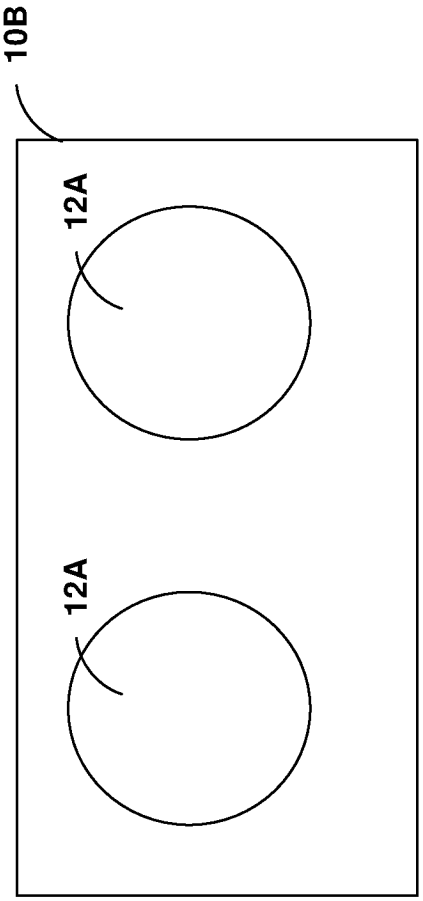


FIG. 1B

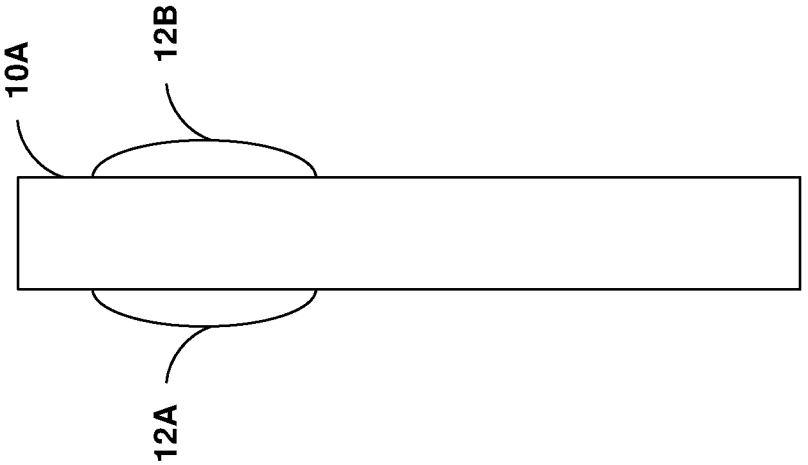


FIG. 1A

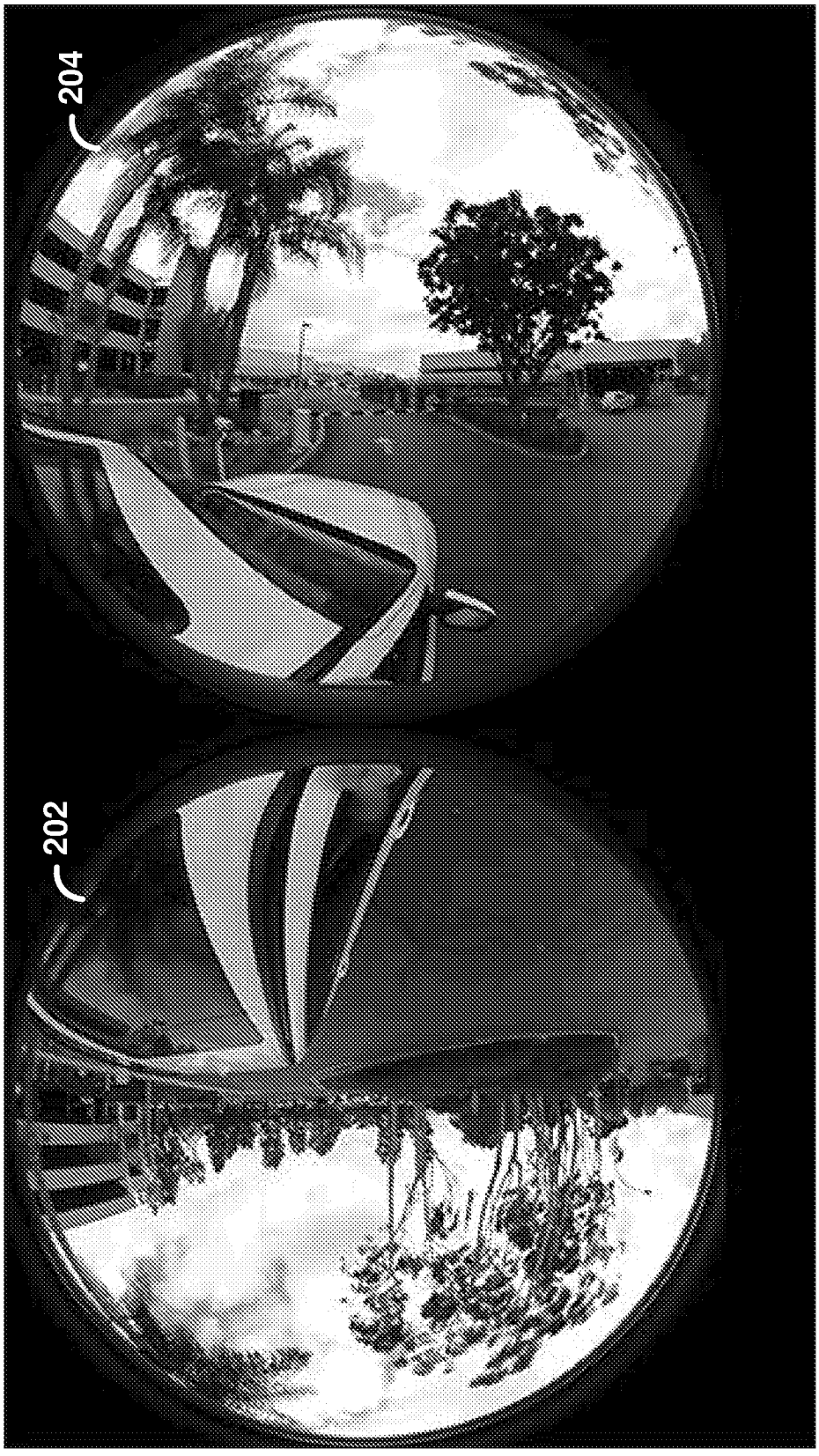


FIG. 2

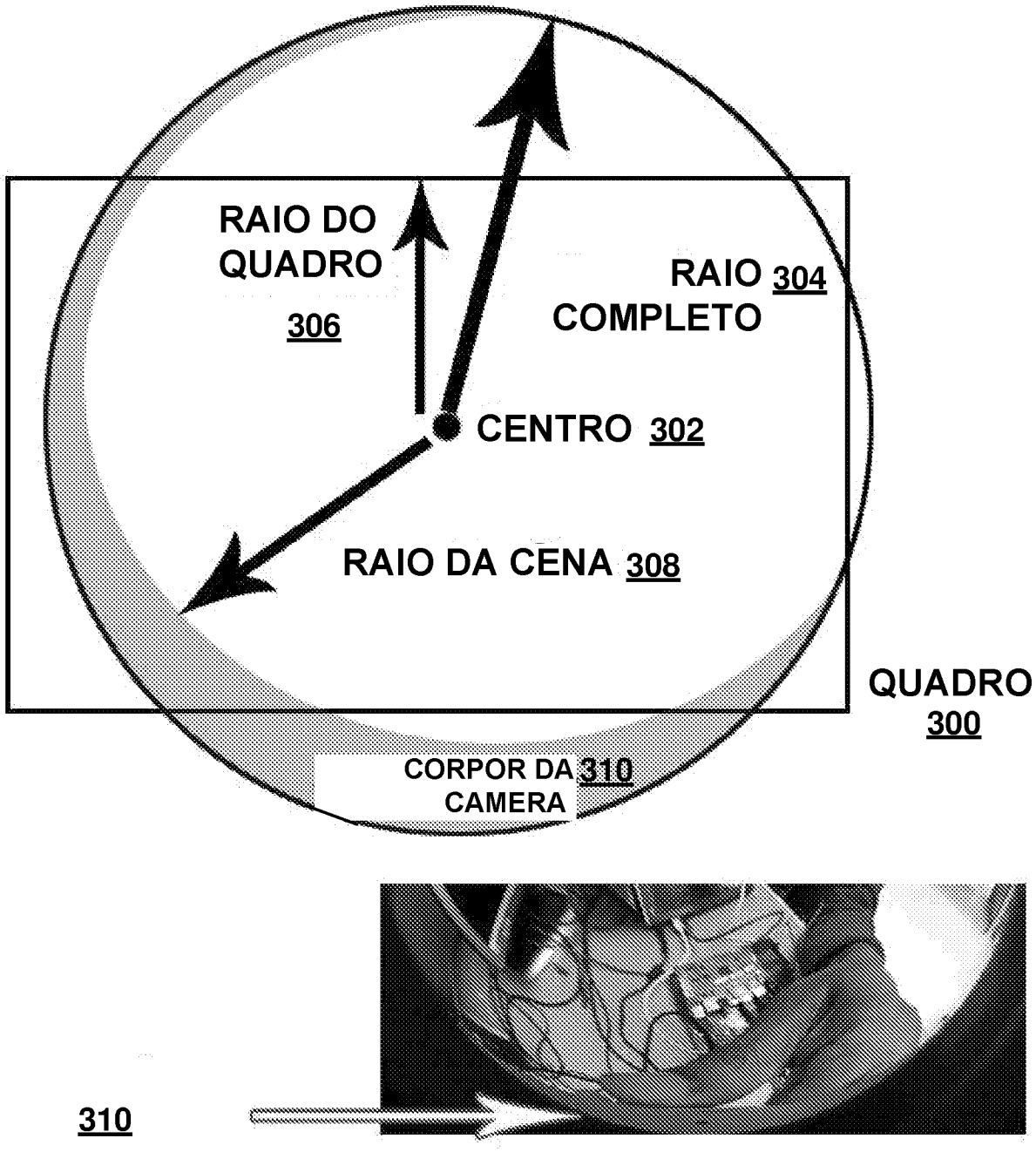

RAIO DO QUADRO 306
RAIO 304 COMPLETO
CENTRO 302
RAIO DA CENA 308
QUADRO 300
CORPOR DA CAMERA 310
310

FIG. 3

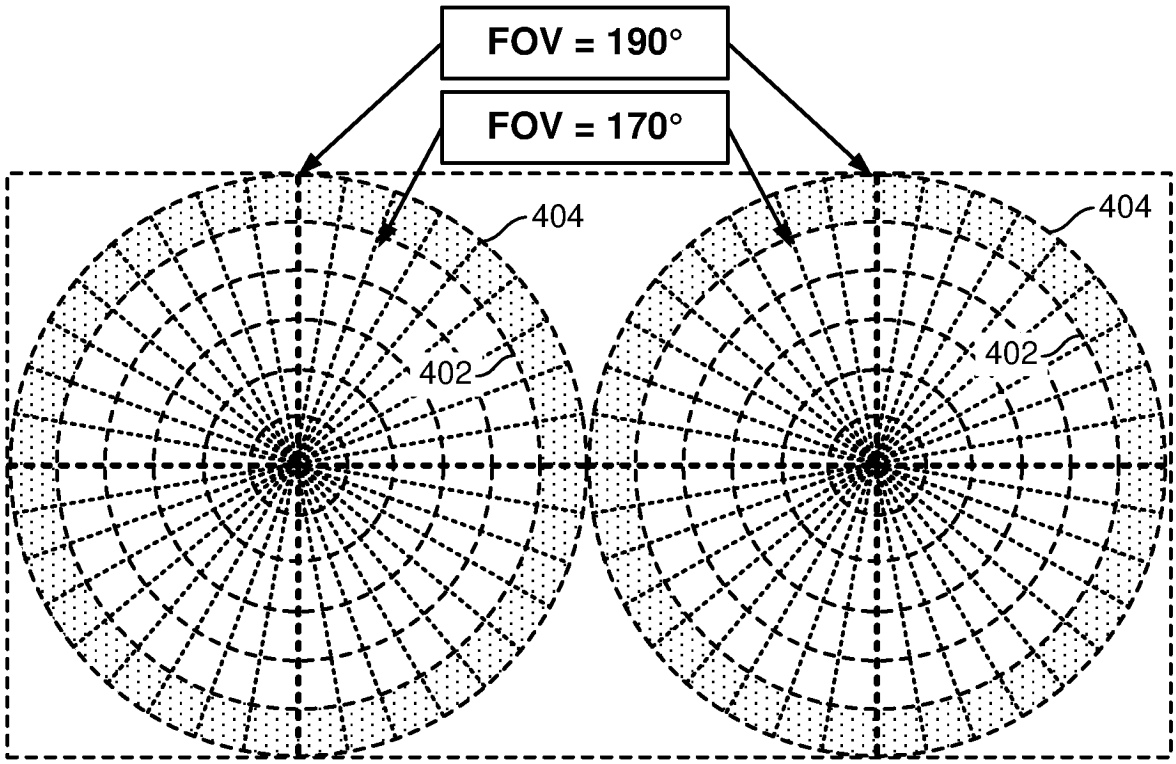


FIG. 4

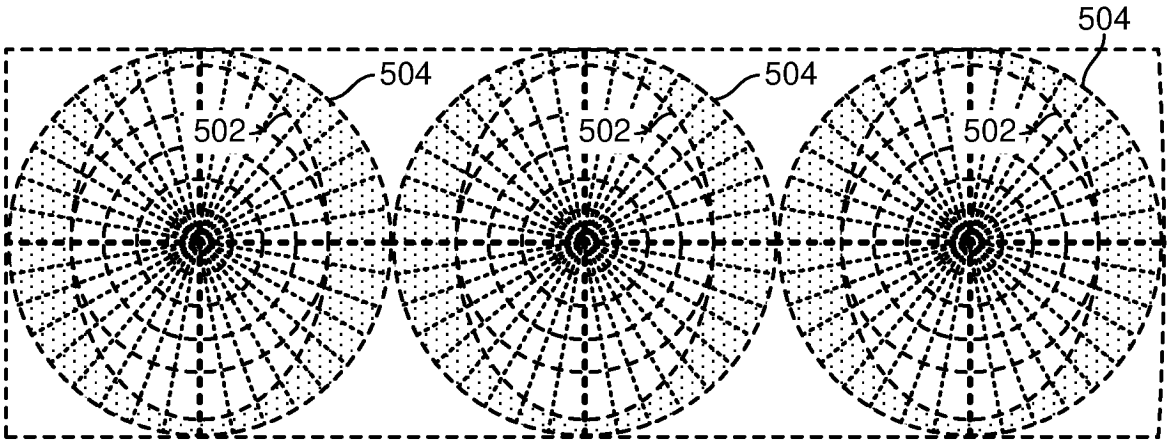
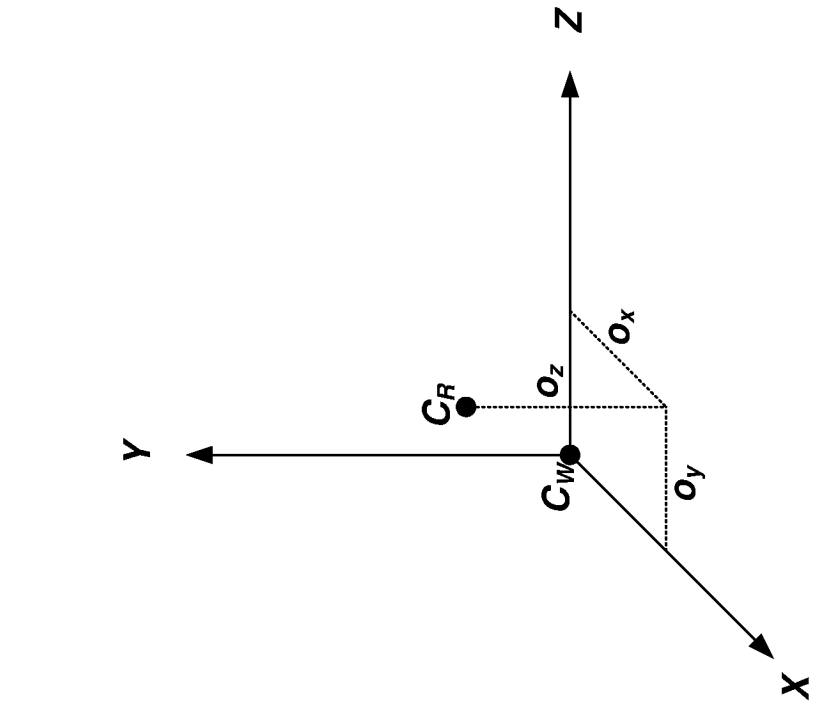


FIG. 5


Z
Y
X
C_R
C_W
o_z
o_x
o_y

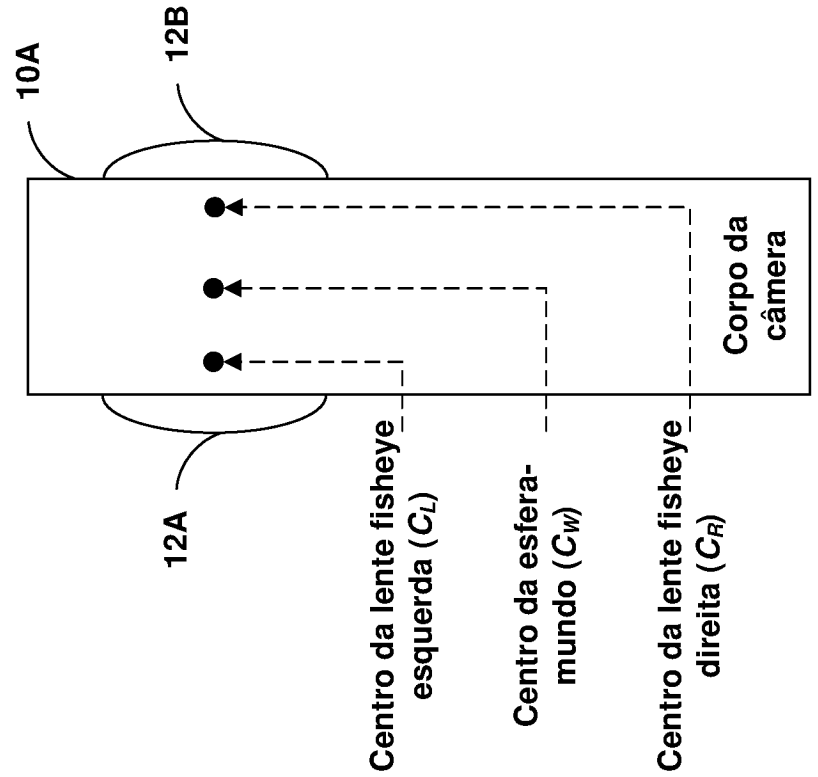
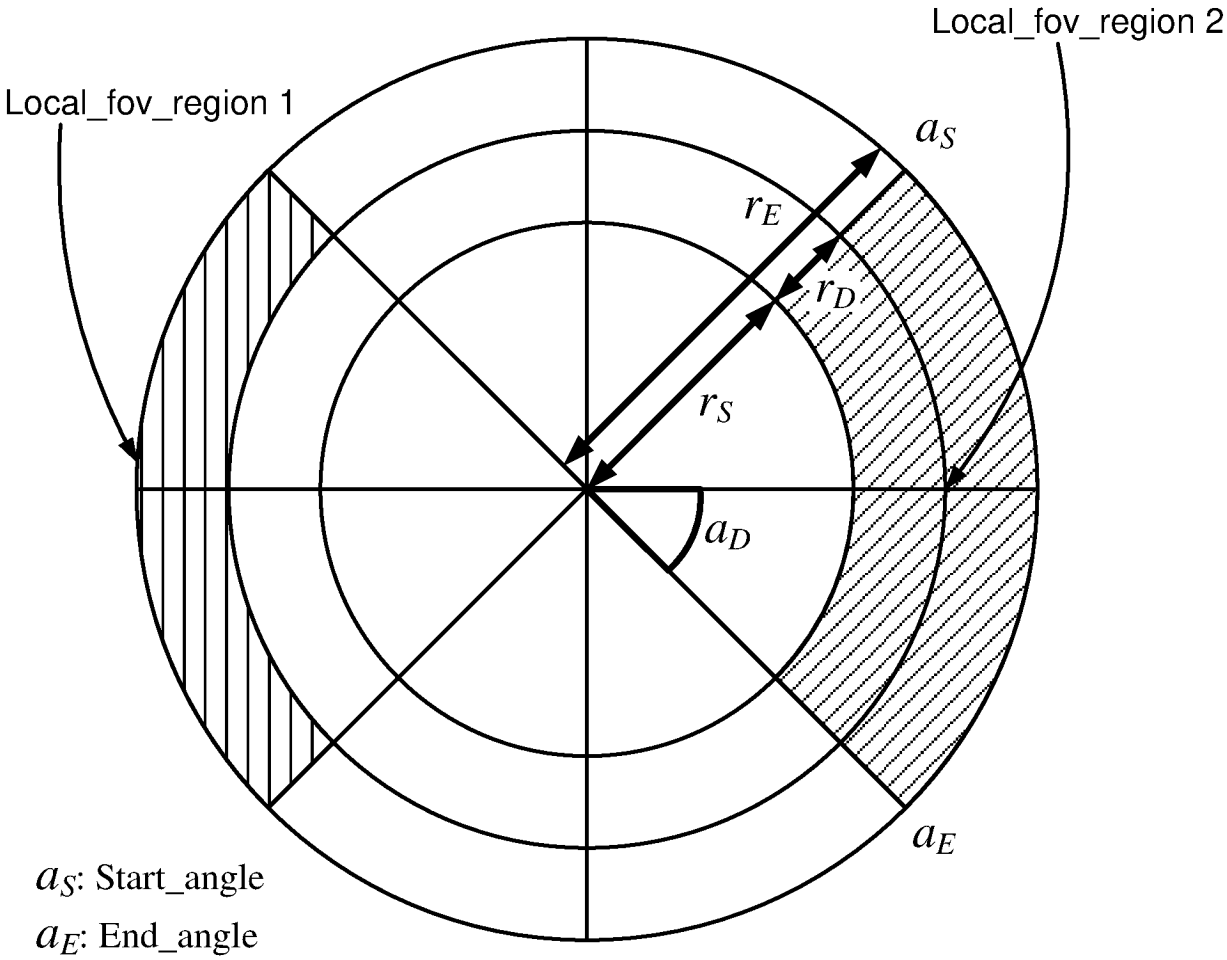

10A
12B
12A
Corpo da câmera
Centro da lente fisheye esquerda (C_L)
Centro da esfera-mundo (C_W)
Centro da lente fisheye direita (C_R)

FIG. 6



a_S: Start_angle
a_E: End_angle
a_D: angle_delta
r_S: Start_radius
r_E: End_radius
r_D: radius_delta

FIG. 7

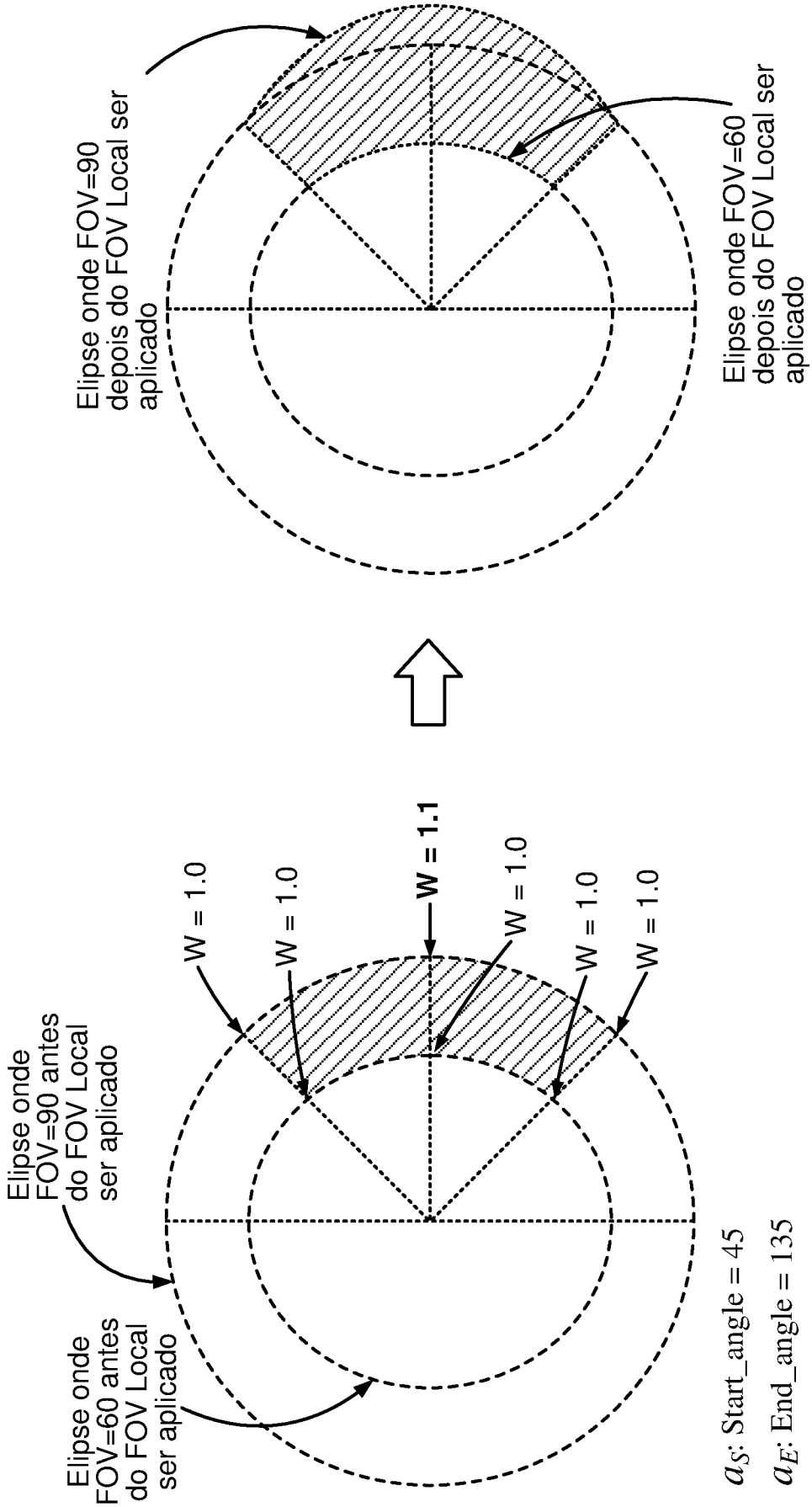

Elipse onde FOV=90 depois do FOV Local ser aplicado
Elipse onde FOV=60 depois do FOV Local ser aplicado
Elipse onde FOV=90 antes do FOV Local ser aplicado
Elipse onde FOV=60 antes do FOV Local ser aplicado
W = 1.0
W = 1.0
W = 1.1
W = 1.0
W = 1.0
W = 1.0
a_S: Start_angle = 45
a_E: End_angle = 135
a_D: angle_delta = 45
r_S: Start_radius = $0.5R_{max}$
r_E: End_radius = $1.0R_{max}$
r_D: radius_delta = $0.5R_{max}$

FIG. 8

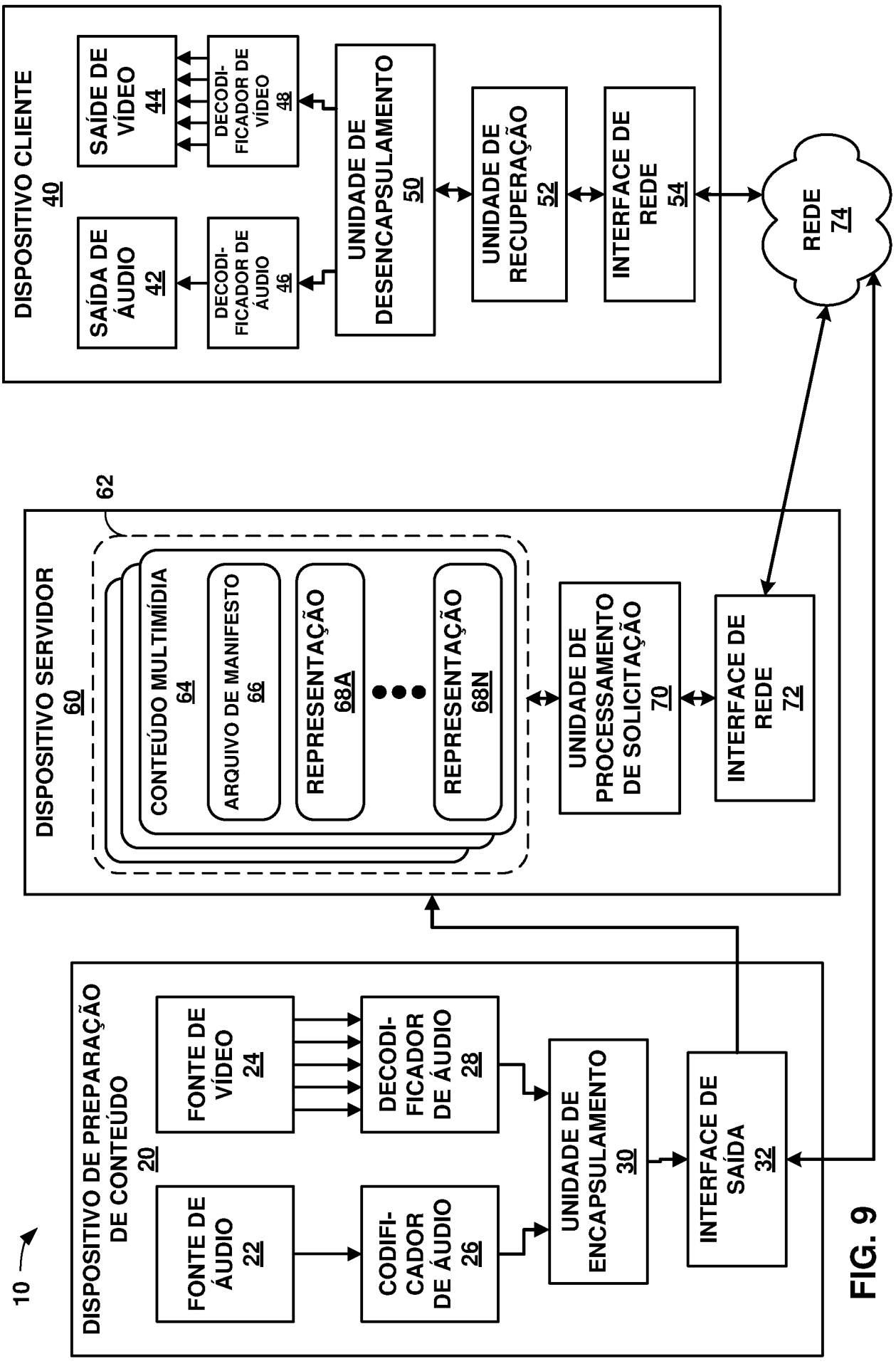

10
DISPOSITIVO DE PREPARAÇÃO DE CONTEÚDO
20
FONTE DE ÁUDIO
22
FONTE DE VÍDEO
24
CODIFICADOR DE ÁUDIO
26
DECODIFICADOR DE ÁUDIO
28
UNIDADE DE ENCAPSULAMENTO
30
INTERFACE DE SAÍDA
32
DISPOSITIVO SERVIDOR
60
62
CONTEÚDO MULTIMÍDIA
64
ARQUIVO DE MANIFESTO
66
REPRESENTAÇÃO
68A
REPRESENTAÇÃO
68N
UNIDADE DE PROCESSAMENTO DE SOLICITAÇÃO
70
INTERFACE DE REDE
72
REDE
74
DISPOSITIVO CLIENTE
40
SAÍDA DE ÁUDIO
42
SAÍDE DE VÍDEO
44
DECODIFICADOR DE ÁUDIO
46
DECODIFICADOR DE VÍDEO
48
UNIDADE DE DESENCAPSULAMENTO
50
UNIDADE DE RECUPERAÇÃO
52
INTERFACE DE REDE
54

FIG. 9

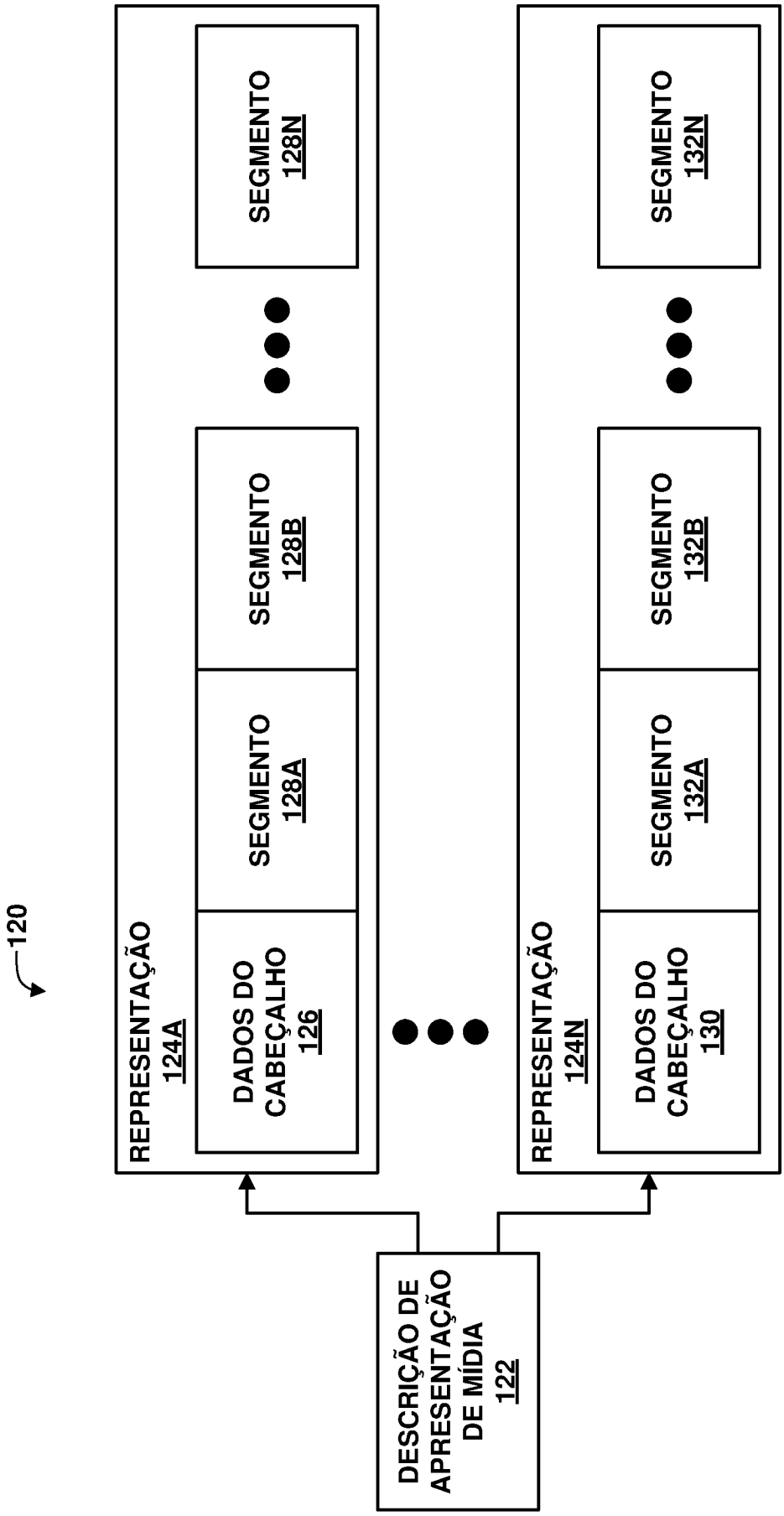

120
DESCRIÇÃO DE APRESENTAÇÃO DE MÍDIA 122
REPRESENTAÇÃO 124A
DADOS DO CABEÇALHO 126
SEGMENTO 128A
SEGMENTO 128B
SEGMENTO 128N
REPRESENTAÇÃO 124N
DADOS DO CABEÇALHO 130
SEGMENTO 132A
SEGMENTO 132B
SEGMENTO 132N

FIG. 10

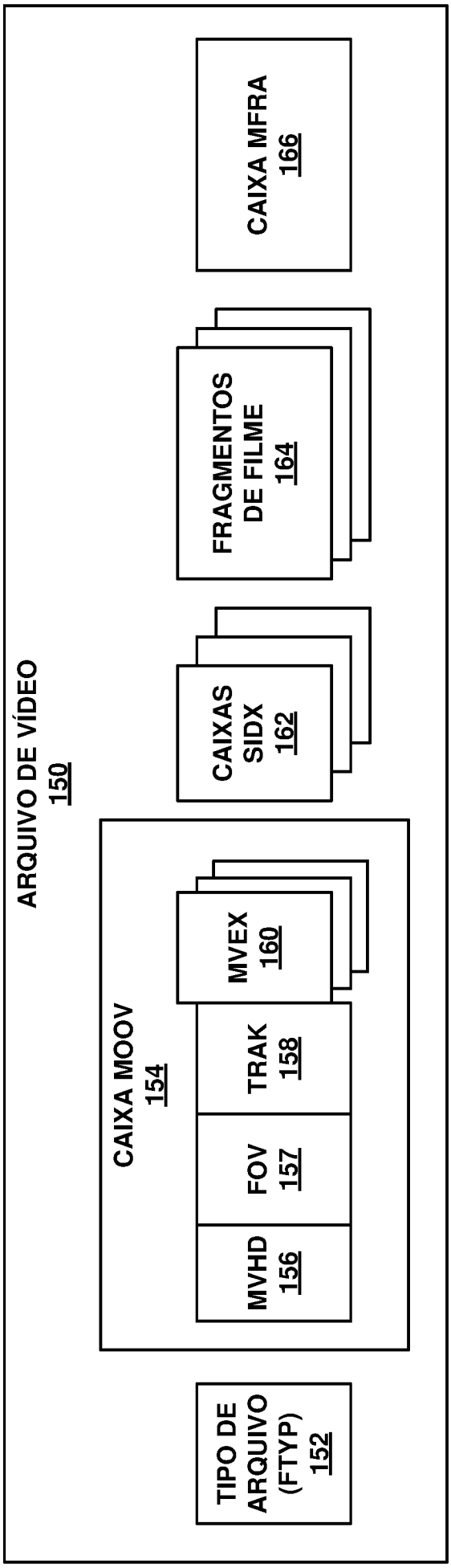

ARQUIVO DE VÍDEO
150
TIPO DE ARQUIVO (FTYP)
152
CAIXA MOOV
154
MVHD
156
FOV
157
TRAK
158
MVEX
160
CAIXAS SIDX
162
FRAGMENTOS DE FILME
164
CAIXA MFRA
166

FIG. 11

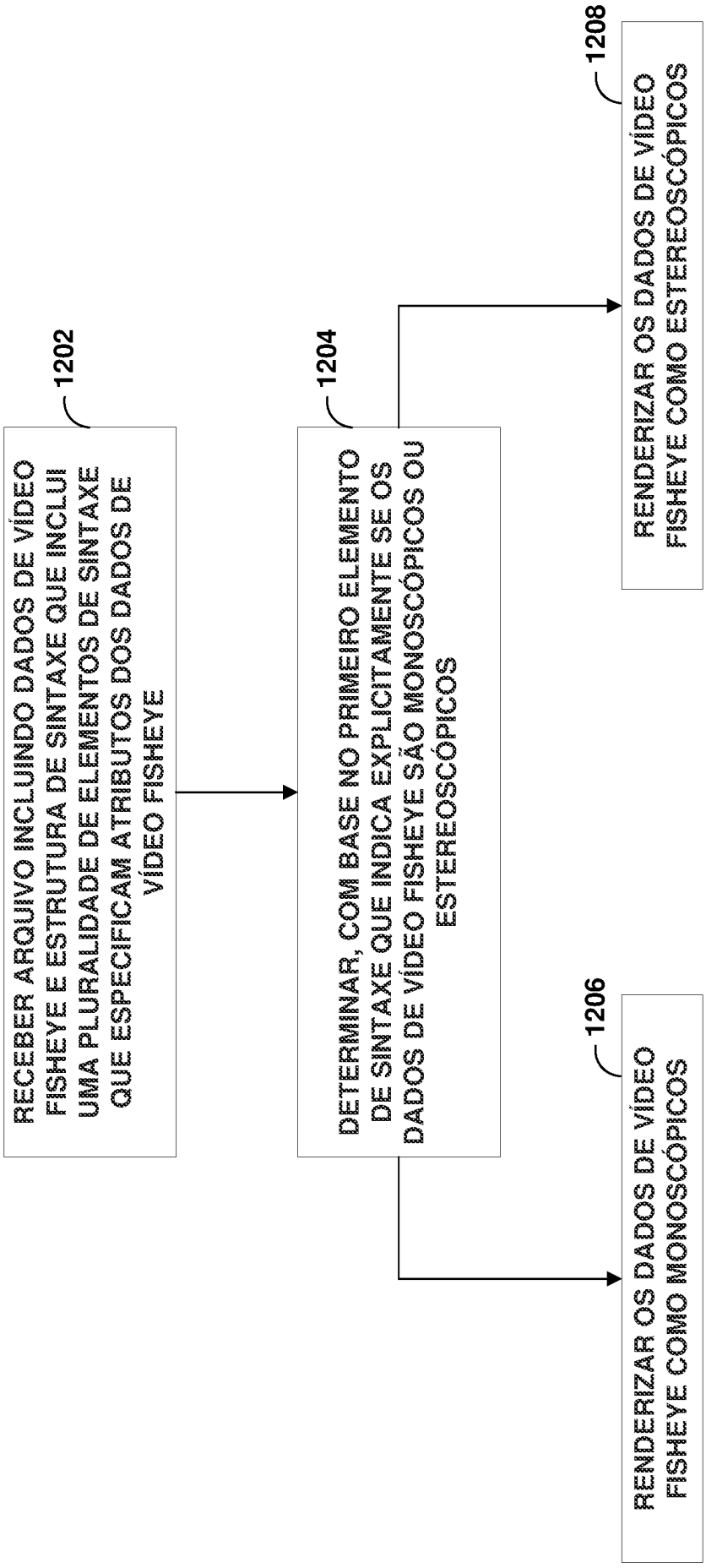

1202
RECEBER ARQUIVO INCLUINDO DADOS DE VÍDEO FISHEYE E ESTRUTURA DE SINTAXE QUE INCLUI UMA PLURALIDADE DE ELEMENTOS DE SINTAXE QUE ESPECIFICAM ATRIBUTOS DOS DADOS DE VÍDEO FISHEYE
1204
DETERMINAR, COM BASE NO PRIMEIRO ELEMENTO DE SINTAXE QUE INDICA EXPLICITAMENTE SE OS DADOS DE VÍDEO FISHEYE SÃO MONOSCÓPICOS OU ESTEREOSCÓPICOS
1206
RENDERIZAR OS DADOS DE VÍDEO FISHEYE COMO MONOSCÓPICOS
1208
RENDERIZAR OS DADOS DE VÍDEO FISHEYE COMO ESTEREOSCÓPICOS

FIG. 12

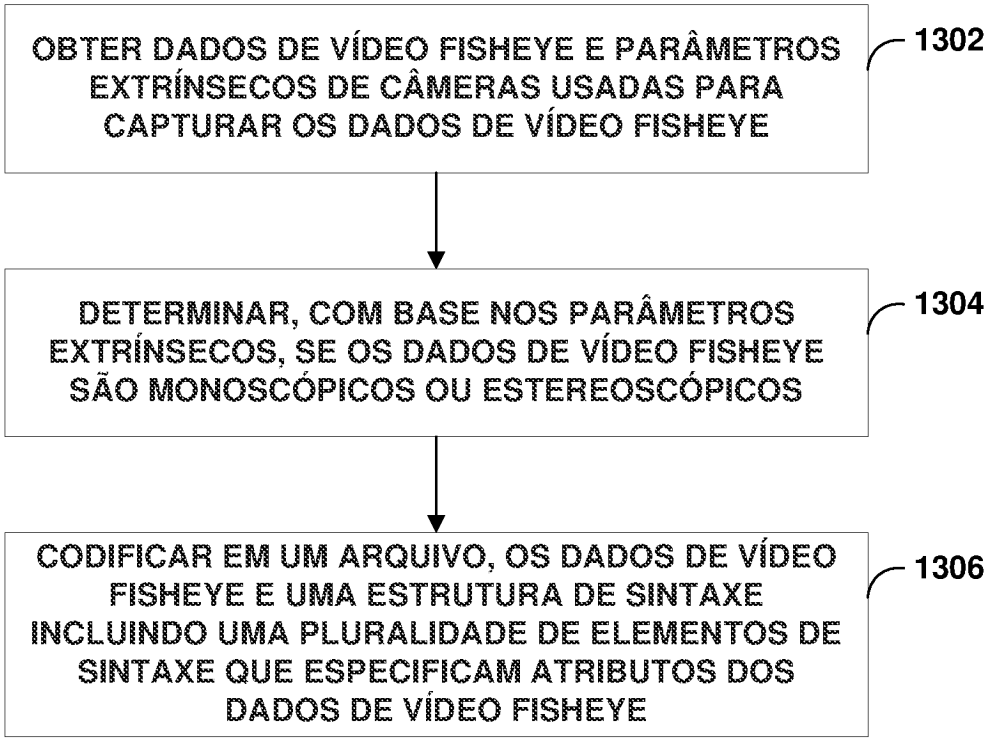

OBTER DADOS DE VÍDEO FISHEYE E PARÂMETROS EXTRÍNSECOS DE CÂMERAS USADAS PARA CAPTURAR OS DADOS DE VÍDEO FISHEYE
1302
DETERMINAR, COM BASE NOS PARÂMETROS EXTRÍNSECOS, SE OS DADOS DE VÍDEO FISHEYE SÃO MONOSCÓPICOS OU ESTEREOSCÓPICOS
1304
CODIFICAR EM UM ARQUIVO, OS DADOS DE VÍDEO FISHEYE E UMA ESTRUTURA DE SINTAXE INCLUINDO UMA PLURALIDADE DE ELEMENTOS DE SINTAXE QUE ESPECIFICAM ATRIBUTOS DOS DADOS DE VÍDEO FISHEYE
1306

FIG. 13

RESUMO

"SINALIZAÇÃO DE ALTO NÍVEL PARA DADOS DE VÍDEO FISHEYE"

Um exemplo de método inclui o processamento de um arquivo incluindo dados de vídeo fisheye, o arquivo incluindo uma estrutura de sintaxe incluindo uma pluralidade de elementos de sintaxe que especificam atributos dos dados de vídeo fisheye, em que a pluralidade de elementos de sintaxe inclui: um primeiro elemento de sintaxe que indica explicitamente se os dados de vídeo fisheye são monoscópicos ou estereoscópicos e um ou mais elementos de sintaxe que indicam implicitamente se os dados do vídeo fisheye são monoscópicos ou estereoscópicos; determinar, com base no primeiro elemento de sintaxe, se os dados de vídeo de fisheye são monoscópicos ou estereoscópicos; e renderizar, com base na determinação, os dados de vídeo fisheye como monoscópicos ou estereoscópicos.