



República Federativa do Brasil

Ministério do Desenvolvimento, Indústria,  
Comércio e Serviços

Instituto Nacional da Propriedade Industrial



**(11) BR 112018014012-0 B1**

**(22) Data do Depósito:** 02/12/2016

**(45) Data de Concessão:** 07/11/2023

**(54) Título:** SISTEMAS E MÉTODOS PARA RENDERIZAR MÚLTIPLOS NÍVEIS DE DETALHE E MEMÓRIA LEGÍVEL POR COMPUTADOR

**(51) Int.Cl.:** G06T 11/40; G06T 15/00; G06T 15/50; G06T 17/20; G06T 3/00; (...).

**(52) CPC:** G06T 11/40; G06T 15/005; G06T 15/503; G06T 17/20; G06T 3/0012; (...).

**(30) Prioridade Unionista:** 12/01/2016 US 14/993,760.

**(73) Titular(es):** QUALCOMM INCORPORATED.

**(72) Inventor(es):** PHI HUNG NGUYEN; NING BI.

**(86) Pedido PCT:** PCT US2016064769 de 02/12/2016

**(87) Publicação PCT:** WO 2017/123341 de 20/07/2017

**(85) Data do Início da Fase Nacional:** 10/07/2018

**(57) Resumo:** Um aparelho eletrônico é descrito. O dispositivo eletrônico inclui um processador. O processador é configurado para renderizar uma primeira zona de uma imagem. O processador também é configurado para renderizar uma segunda zona da imagem. A primeira zona tem um nível tecelado mais elevado de detalhe do que a segunda zona. O processador é adicionalmente configurado para apresentar a primeira zona e a segunda zona em pelo menos uma janela de veículo.

“SISTEMAS E MÉTODOS PARA RENDERIZAR MÚLTIPLOS NÍVEIS DE DETALHE E MEMÓRIA LEGÍVEL POR COMPUTADOR”

CAMPO DE DIVULGAÇÃO

[0001] A presente invenção refere-se genericamente a dispositivos eletrônicos. Mais especificamente, a presente divulgação refere-se a sistemas e métodos para renderizar múltiplos níveis de detalhe.

FUNDAMENTOS

[0002] Alguns dispositivos eletrônicos (por exemplo, smartphones, computadores, televisores, automóveis, sistemas de videogame, reprodutores de mídia, câmeras, telefones celulares, câmeras pessoais, set-top boxes, projetores, displays, etc.) exibem imagens. Por exemplo, um smartphone pode capturar e/ou processar imagens estáticas e/ou de vídeo. Processamento de imagens pode exigir uma quantidade relativamente grande de recursos de tempo, memória e energia. Os recursos exigidos podem variar de acordo com a complexidade do processamento.

[0003] o processamento de grandes quantidades de dados de imagem pode ser particularmente difícil. Por exemplo, grandes quantidades de dados de imagem podem exigir quantidades relativamente grandes de recursos, como a energia, tempo, largura de banda do processador, memória e/ou custos monetários. Como pode ser observado a partir desta discussão, sistemas e métodos que melhoram o processamento de imagem podem ser benéficos.

SUMÁRIO

[0004] Um dispositivo eletrônico é descrito. O dispositivo eletrônico inclui um processador configurado para renderizar uma primeira zona de uma imagem. O

processador também é configurado para renderizar uma segunda zona da imagem. A primeira zona tem um nível tecelado mais elevado de detalhe do que a segunda zona. O processador é adicionalmente configurado para apresentar a primeira zona e a segunda zona em pelo menos uma janela de veículo. O dispositivo eletrônico pode ser pelo menos parte de um módulo removível em um veículo.

[0005] O dispositivo eletrônico pode incluir pelo menos um display. O processador pode ser configurado para enviar a primeira zona e a segunda zona para o pelo menos um display para apresentar a primeira zona e a segunda zona. O pelo menos um display pode ser um projetor de janela ou de um painel de exibição de janela integrada.

[0006] O processador pode ser configurado para apresentar a primeira zona em uma primeira janela e para apresentar a segunda zona em uma segunda janela. A primeira janela pode ser um para-brisa frontal e a segunda janela pode ser uma outra janela de um veículo.

[0007] O processador pode ser configurado para apresentar a primeira zona em uma primeira seção de uma janela e pode ser configurado para apresentar a segunda zona em uma segunda seção da janela. A primeira seção pode ter uma seção lateral de um para-brisa e a segunda seção pode ser uma outra seção lateral do para-brisa. A primeira seção pode ser uma primeira barra horizontal através de um para-brisa e a segunda seção pode ser uma segunda barra horizontal através do para-brisa.

[0008] O processador pode ser configurado para realizar o rastreamento ocular (eye tracking). O processador pode também ser configurado para renderizar a primeira zona

e a segunda zona baseado em um rastreamento ocular. O processador pode ser configurado para executar rastreamento ocular por múltiplos usuários. O processador pode priorizar renderização de zona baseado no rastreamento ocular.

[0009] O processador pode ser configurado para renderizar uma terceira zona. A terceira zona pode ter um nível de detalhe tecelado inferior do que a primeira zona e a segunda zona. O processador pode ser configurado para apresentar a terceira zona na pelo menos uma janela do veículo.

[0010] Um método realizado por um dispositivo eletrônico também é descrito. O método inclui a renderizar uma primeira zona de uma imagem. O método também inclui renderizar uma segunda zona da imagem. A primeira zona tem um nível tecelado mais elevado de detalhe do que a segunda zona. O método inclui ainda apresentar a primeira zona e a segunda zona em pelo menos uma janela de veículo.

[0011] Um produto de programa de computador é também descrito. O produto de programa de computador inclui um meio legível por computador tangível não transitório com instruções. As instruções incluem código para fazer com que um dispositivo eletrônico renderize uma primeira zona de uma imagem. As instruções incluem também código para fazer com que o dispositivo eletrônico renderize uma segunda zona da imagem. A primeira zona tem um nível tecelado mais elevado de detalhe do que a segunda zona. As instruções incluem ainda código para fazer com que o dispositivo eletrônico apresente a primeira zona e a segunda zona em pelo menos uma janela de veículo.

[0012] Um aparelho também é descrito. O aparelho inclui meios para renderizar uma primeira zona de uma imagem. O aparelho também inclui meios para renderizar uma segunda zona da imagem. A primeira zona tem um nível tecelado mais elevado de detalhe do que a segunda zona. O aparelho inclui ainda meios para apresentar a primeira zona e a segunda zona em pelo menos uma janela de veículo.

#### BREVE DESCRIÇÃO DOS DESENHOS

[0013] A Figura 1 é um diagrama de blocos que ilustra uma configuração de um dispositivo eletrônico, em que os sistemas e métodos para renderizar múltiplos níveis de detalhe podem ser implementados;

[0014] A Figura 2 é um diagrama de fluxo que ilustra uma configuração de um método para produzir múltiplos níveis de detalhe;

[0015] A Figura 3 é um diagrama de blocos que ilustra uma configuração de um dispositivo eletrônico, em que os sistemas e métodos para renderizar múltiplos níveis de detalhe podem ser implementados;

[0016] A Figura 4 é um diagrama que ilustra um exemplo de um buffer (armazenador temporário) de quadro não uniforme;

[0017] A Figura 5 é um diagrama de fluxo que ilustra uma configuração mais específica de um método para produzir múltiplos níveis de detalhe;

[0018] A Figura 6 é um diagrama de fluxo que ilustra exemplo de um método para renderização foveada;

[0019] A Figura 7 ilustra um exemplo de renderização foveada;

[0020] A Figura 8 é um diagrama de blocos que ilustra um exemplo de um dispositivo eletrônico incluindo um circuito de palavra de instrução muita longa (VLIW);

[0021] A Figura 9 é um diagrama de blocos que ilustra um exemplo dos sistemas e processos divulgados aqui implementados na arquitetura descrita em ligação com a Figura 8;

[0022] A Figura 10 é um diagrama de blocos que ilustra um outro exemplo de um dispositivo eletrônico incluindo um circuito de palavra de instrução muita longa (VLIW);

[0023] A Figura 11 é um diagrama de blocos que ilustra um outro exemplo dos sistemas e processos divulgados aqui implementados na arquitetura descrita em ligação com a Figura 10;

[0024] A Figura 12 é um diagrama de blocos que ilustra um outro exemplo de um dispositivo eletrônico incluindo um circuito de palavra de instrução muita longa (VLIW);

[0025] A Figura 13 é um diagrama de blocos que ilustra um exemplo dos sistemas e processos divulgados aqui implementados na arquitetura descrita em ligação com a Figura 12;

[0026] A Figura 14 é um diagrama de blocos que ilustra um outro exemplo de um dispositivo eletrônico incluindo um circuito de palavra de instrução muita longa (VLIW);

[0027] A Figura 15 é um diagrama de blocos que ilustra um exemplo dos sistemas e processos divulgados aqui

implementados na arquitetura descrita em ligação com a Figura 14;

[0028] A Figura 16 é um diagrama de blocos que ilustra uma configuração de um dispositivo eletrônico, em que os sistemas e métodos para renderizar várias zonas podem ser implementados;

[0029] A Figura 17 é um diagrama de fluxo que ilustra um método para renderizar várias zonas;

[0030] A Figura 18 é um diagrama que ilustra alguns exemplos de displays de janelas;

[0031] A Figura 19 é um diagrama que ilustra um exemplo de zonas sobre uma janela;

[0032] A Figura 20 é um diagrama que ilustra um outro exemplo de zonas sobre uma janela;

[0033] A Figura 21 é um diagrama que ilustra um outro exemplo de zonas sobre múltiplas janelas;

[0034] A Figura 22 é um diagrama de fluxo que ilustra uma configuração mais específica de um método para processamento de múltiplas zonas;

[0035] A Figura 23 é um diagrama que ilustra um outro exemplo de zonas sobre uma janela; e

[0036] A Figura 24 ilustra certos componentes que podem ser incluídos dentro de um dispositivo eletrônico.

#### DESCRIÇÃO DETALHADA

[0037] Os sistemas e métodos aqui descritos podem prover abordagens para renderizar múltiplos níveis de detalhe. Por exemplo, múltiplos diferentes níveis de detalhe podem ser renderizados na mesma imagem. Isso pode melhorar a eficiência de renderização e/ou experiência do usuário. Tal como aqui utilizado, o termo "renderizar" e as suas

variações podem significar realizar uma ou mais operações para a produção de dados de imagem. Por exemplo, renderização pode incluir a realização de uma ou mais operações para converter dados geométricos (por exemplo, modelos geométricos, vértices, polígonos, etc.) para os dados de imagem (por exemplo, dados de pixel, pixels, dados de domínio visuais, etc.). Os dados de imagem resultantes de processamento podem ser apresentados em um display, armazenados e/ou transmitidos. Por exemplo, dados de imagem renderizados podem ou não ser apresentados em um display imediatamente. Os dados de imagem renderizada podem ser armazenados (para apresentação posterior em um display, por exemplo), podem ser transmitidos para outro dispositivo (que pode apresentar os dados de imagem renderizada em um display, por exemplo) e/ou podem ser apresentados em um display.

[0038] Ineficiência de renderização pode ser um problema atual. Por exemplo, a realidade virtual (VR) pode incluir a renderização de duas visualizações de uma cena, o que pode renderizar o conteúdo duas vezes mais lento para renderizar. Uma abordagem pode envolver a realização de um rastreamento ocular (que pode ser realizado com precisão e baixo custo), renderizar três passagens geometrias com diferentes níveis de rudeza, realizando o antialiasing e mesclando as três diferentes qualidades de buffer de quadro. No entanto, essa abordagem podem ser ineficientes, uma vez que as várias passagens de geometria podem se sobrepor, resultando em alguns pixels sendo processados repetidamente.

[0039] Em algumas configurações, os sistemas e métodos aqui descritos podem prover abordagens multinível de única passagem de renderização de detalhe (multi-LOD) (por

exemplo, renderização foveada de única passagem). Por exemplo, algumas abordagens podem utilizar um tessellator (por exemplo, máquina de tecelagem), antialiasing multiamostra (MSAA), envio de núcleo dinâmico e/ou um buffer de quadro multirresolução (por exemplo, não uniforme) (por exemplo, objeto de buffer de quadro (FBO)). Em algumas abordagens, pode ser necessária apenas uma passagem de geometria. Esta melhoria pode reduzir o uso de largura de banda de geometria e/ou pode reutilizar MSAA.

[0040] As configurações de alguns dos sistemas e métodos aqui divulgados podem oferecer um ou mais benefícios de desempenho. Por exemplo, hardware pode ser implementado, que inclui um FBO multirresolução e/ou envio de núcleo dinâmico. Isto pode reduzir o consumo de largura de banda. Por exemplo, múltiplas passagens de geometria (por exemplo, três) podem ser colapsadas em uma usando o tessellator (por exemplo, máquina de tecelagem). Antialiasing de hardware de rasterização pode ser reutilizado em algumas configurações. Isto pode permitir melhorias na renderização de conteúdo regular. Adicionalmente ou em alternativa, envio dinâmico de núcleos de fragmento pode reduzir computação. Estas melhorias podem ser particularmente benéficas no contexto da realidade virtual (VR) e/ou realidade aumentada (AR), onde múltiplas visualizações de uma cena podem ser renderizadas de cada vez (para visualizações estereoscópicas, por exemplo). Em algumas configurações, RV e RA podem ser combinadas. Os sistemas e métodos aqui divulgados podem ser benéficos, como a indústria de processamento gráfico buscando por maneiras de prover experiências imersivas.

[0041] Os sistemas e métodos aqui descritos também descrevem abordagens para renderizar múltiplas zonas para apresentação em janelas. Estes sistemas e métodos podem ser particularmente benéficos no contexto de veículos para melhorar a experiência de usuário em aplicações tais como a ajuda de condutor e veículos autônomos (por exemplo, carros de autocondução).

[0042] Várias configurações são agora descritas com referência às Figuras, em que números de referência semelhantes, podem indicar elementos funcionalmente semelhantes. Os sistemas e métodos geralmente como descritos e ilustrados nas figuras aqui podem ser dispostos e concebidos de uma ampla variedade de configurações diferentes. Assim, a seguinte descrição mais detalhada de várias configurações, como representado nas figuras, não se destina a limitar o escopo, tal como reivindicado, mas é meramente representativo dos sistemas e métodos.

[0043] A Figura 1 é um diagrama de blocos que ilustra uma configuração de um dispositivo eletrônico 102 no qual os sistemas e métodos para renderizar múltiplos níveis de detalhe podem ser implementados. Exemplos do dispositivo eletrônico 102 incluem smartphones, computadores (por exemplo, desktops, laptops, etc.), displays head-mounted (por exemplo, fones de ouvido de realidade Virtual (RV), fones de ouvido de realidade aumentada (AR), etc.), dispositivos VR, dispositivos AR, tablets, reprodutores de mídia, televisores, automóveis, veículos, máquinas fotográficas, câmeras de vídeo, câmeras digitais, telefones celulares, câmeras pessoais, câmeras de ação, aeronaves, aviões, veículos aéreos não tripulados (UAVs), equipamentos

de saúde, consoles de jogos, assistentes digitais pessoais (PDAs), set-top boxes, leitores de mídia, etc. O dispositivo eletrônico 102 pode incluir um ou mais componentes ou elementos. Um ou mais dos componentes ou elementos podem ser implementados em hardware (por exemplo, circuitos), firmware e/ou uma combinação de hardware e software (por exemplo, um processador com instruções).

[0044] Em algumas configurações, o dispositivo eletrônico 102 pode incluir um circuito de palavra de instrução muita longa (VLIW) 104 e a memória de sistema 114. O circuito VLIW 104 pode ser um circuito (por exemplo, processador) que implementa paralelismo de nível de instruções. Por exemplo, o circuito VLIW 104 pode executar instruções em paralelo (por exemplo, ao mesmo tempo). O circuito VLIW 104 pode ser acoplado a (por exemplo, em comunicação eletrônica com) a memória de sistema 114. O circuito VLIW 104 pode ser um dispositivo de renderização de imagens. Por exemplo, o circuito VLIW 104 pode converter dados de geometria (por exemplo, dados de geometria de modelo (s) geométrico, bidimensional (2D) ou tridimensional (3D), vértices, polígonos, etc.) em uma ou mais imagens. A imagem (s) resultante pode ser apresentada em um ou mais displays. Por exemplo, a imagem (s) pode ser uma única imagem para produzir uma visualização 2D ou pode ser múltiplas imagens (por exemplo, imagens estereoscópicas) para produzir uma visualização 3D. Por exemplo, o circuito VLIW 104 pode produzir imagens múltiplas (por exemplo, diferentes visualizações) para uma visualização estereoscópica (por exemplo, uma imagem de um olho esquerdo e outro para uma imagem do olho direito). Deve ser notado que o circuito VLIW

104 pode renderizar uma série de imagens (por exemplo, vídeo). O circuito VLIW 104 pode ser implementado em hardware ou em uma combinação de hardware e software. Um exemplo do circuito VLIW 104 é uma unidade de processamento gráfico (GPU).

[0045] Em algumas configurações, o dispositivo eletrônico 102 pode executar uma ou mais das funções, processos, métodos, etapas, etc. descritos em ligação com uma ou mais das Figuras 2-5, 9, 11, 13 e 15-24. Adicionalmente ou em alternativa, o dispositivo eletrônico 102 pode incluir uma ou mais das estruturas descritas em ligação com uma ou mais das Figuras 2-5, 9, 11, 13 e 15-24.

[0046] A memória de sistema 114 pode armazenar instruções e/ou dados. Exemplos de memória de sistema 114 inclui a memória de acesso aleatório (RAM), memória de acesso aleatório dinâmica (DRAM), memória de acesso aleatório dinâmica síncrona (SDRAM), memória de acesso aleatório dinâmica síncrona de taxa de dados dupla (DDR SDRAM), DDR2, DDR3, DDR4, memória de acesso aleatório estática (SRAM), registradores, memória somente de leitura (ROM), memória somente de leitura programável apagável (EPROM), memória somente de leitura programável eletricamente apagável (EEPROM), memória flash NAND, memória flash NOR, etc. O circuito VLIW 104 pode acessar (por exemplo, ler a partir de e/ou gravar em) a memória de sistema 114. A memória de sistema 114 pode ser acessível por outros elementos (se houver) do dispositivo eletrônico 102, tal como um ou mais processadores (por exemplo, unidades de processamento central (CPUs)). A memória de sistema 114 pode ser separada do circuito VLIW 104. Exemplos de instruções e/ou dados que

podem ser armazenados pelo memória de sistema 114 podem incluir instruções, dados de comando, dados geométricos (por exemplo, vértices, polígonos, normais (por exemplo, orientações de objeto), primitivas, metadados, etc.), dados de textura, dados de visualização de acesso não ordenados, parâmetro (s) de renderização (por exemplo, parâmetro de nível de detalhe (s)), etc. Em algumas configurações, o dispositivo eletrônico 102 (por exemplo, o memória de sistema 114) pode incluir um ou mais buffers de quadro. O buffer (s) de quadro pode armazenar temporariamente (por exemplo, armazenar) dados de imagem. Os dados de imagem em buffer podem ser apresentados em um ou mais displays.

[0047] O circuito VLIW 104 pode incluir um ou mais elementos de renderizar imagens. Alguns exemplos dos elementos podem incluir um front end (EF), processador de vértice (VPS, coletor de vértice (VF), shader de vértice (VS), shader de controle de tecelagem (TCS), hull shader (HS), tessellator (por exemplo, máquina de tecelagem) (TESS), shader de avaliação de tecelagem (TES), shader de domínio (DS), shader de geometria (GS), rasterizador (RAST), shader de fragmento (FS), pixel shader (PS), mosaico, unidade de saída de renderização (ROP), clipper (clip) e/ou saída de fluxo (SOL), etc. Em algumas configurações, um ou mais destes elementos podem ser referidos como um pipeline gráfico 106. Deve ser notado que o tessellator pode ser uma máquina de tecelagem de função fixa em alguns configurações. O tessellator (e/ou um ou mais outros elementos de um pipeline gráfico) pode ser implementado em hardware, software ou uma combinação de hardware e software

[0048] O circuito VLIW 104 pode incluir memória em chip 110. A memória em chip 110 pode ser implementada no circuito VLIW 104 para prover acesso à memória rápida. Por exemplo, a memória em chip 110 pode ser acessada de forma mais rápida do que a memória de sistema 114. Exemplos de memória em chip 110 podem incluir memória de acesso aleatório (RAM), memória de acesso aleatório dinâmica (DRAM), memória de acesso aleatório dinâmica síncrona (SDRAM), memória de acesso aleatório dinâmica síncrona de taxa de dados dupla (DDR SDRAM), DDR2, DDR3, DDR4, memória de acesso aleatório estática (SRAM), registradores, memória somente de leitura programável (ROM), memória somente de leitura programável apagável (EPROM), memória somente de leitura programável eletricamente apagável (EEPROM), memória flash NAND, memória flash NOR, etc. Em algumas configurações, a memória em chip 110 pode ser implementada em múltiplos níveis (por exemplo, cache de nível 1 (L1), cache de nível 2 (L2), cache de nível 3 (L3), etc.). Exemplos de dados que podem ser armazenados pela memória em chip 110 podem incluir dados de geometria (por exemplo, os vértices, polígonos, normais, primitivas, metadados, etc.), dados de textura, dados de visualização de acesso não ordenados, um ou mais parâmetros de renderização (por exemplo, parâmetro (s) de nível de detalhe (LOD)), etc. Em algumas configurações, a memória em chip 110 pode incluir um ou mais buffer (s) de quadro (além de ou em alternativa a partir da memória de sistema 114, por exemplo). O buffer (s) de quadro pode armazenar em buffer (por exemplo, armazenar) dados de imagem. Os dados de imagem em buffer podem ser apresentados em um ou mais displays.

[0049] Em algumas configurações, o circuito VLIW 104 pode incluir um controlador de memória assíncrona 108. O controlador de memória assíncrona 108 pode ser implementado em hardware ou em uma combinação de hardware e software. O controlador de memória assíncrona 108 pode ser implementado no circuito VLIW 104, separadamente do circuito VLIW, ou pode ser implementado parcialmente no circuito VLIW 104 e parcialmente separado do circuito VLIW 104. O controlador de memória assíncrona 108 pode ser configurado para acessar a memória (por exemplo, memória em chip 110 e/ou a memória de sistema 114) de forma assíncrona. Em algumas configurações, o controlador de memória assíncrona 108 opera "de forma assíncrona" no sentido de que o controlador de memória assíncrona 108 pode acessar a memória (por exemplo, parâmetro (s) de nível de detalhe (LOD), etc.) fora de sincronização com um conjunto de dados que está a ser processado (por exemplo, tecelado). Por exemplo, o circuito VLIW 104 pode periodicamente processar conjuntos de dados. Processar (por exemplo, tecelar) um único conjunto de dados pode ser referido como uma chamada de desenho. Por exemplo, uma chamada de desenho pode ser uma instrução (por exemplo, uma solicitação de programação, a instrução, etc.) para o circuito VLIW 104 processar (por exemplo, renderizar) um conjunto de dados. Um ou mais parâmetros (por exemplo, parâmetro de LOD, configuração de tecelagem, um ou mais de outros parâmetros, etc.) podem ser associados com e/ou definidos para a chamada de desenho. Por conseguinte, uma chamada de desenho pode estabelecer um LOD a ser renderizado para o conjunto de dados, por exemplo.

[0050] Em abordagens conhecidas, cada conjunto de dados (por exemplo, chamada de desenho) tem um nível fixo de detalhe (LOD). Nestas abordagens, o LOD para um conjunto de dados não pode ser alterado, uma vez que processamento já começou naquele conjunto de dados (por exemplo, não pode mudar durante uma chamada de desenho). Além disso, cada conjunto de dados pode ter apenas um único LOD em abordagens conhecidas.

[0051] De acordo com os sistemas e métodos aqui descritos, o controlador de memória assíncrona 108 pode de forma assíncrona acessar a memória (por exemplo, memória em chip 110 e/ou a memória de sistema 114) para permitir renderizar diferentes níveis de detalhe (por exemplo, zonas diferentes com diferentes níveis de detalhe) dentro de um único conjunto de dados (por exemplo, durante o processamento de uma chamada de desenho). Por exemplo, o controlador de memória assíncrona 108 pode acessar a memória (por exemplo, o parâmetro (s), parâmetro de LOD (s), etc.) após o processamento (por exemplo, tecelagem, shading, etc.) em um conjunto de dados que já começou e antes do processamento do conjunto de dados ser concluído. Isto pode permitir que o circuito VLIW 104 altere LOD durante o processamento de um conjunto de dados para produzir diferentes níveis de detalhe em um conjunto de dados. Cada conjunto de dados pode incluir múltiplos primitivos (por exemplo, pontos, polígonos, etc.). Em algumas configurações, cada conjunto de dados pode (inicialmente) ter um único LOD atribuído. No entanto, o controlador de memória assíncrona 108 pode atribuir um nível de detalhe diferente para um conjunto de dados durante o processamento desse conjunto de dados. Em outras palavras,

o controlador de memória assíncrona 108 pode alterar o LOD para um conjunto de dados (depois do processamento (por exemplo, tecelagem, shading, etc.) sobre aquele conjunto de dados que já começou, mas antes do processamento (por exemplo, tecelagem, shading, etc.) sobre aquele conjunto de dados que foi concluído). Por exemplo, os sistemas e métodos aqui divulgados podem permitir a mudança de LOD em uma base por pixel e/ou per-primitivo (mesmo dentro de um conjunto de dados, por exemplo). Em algumas configurações, os parâmetros (por exemplo, o parâmetro (s), parâmetro de LOD (s), etc.) podem ser atualizados durante o processamento. Por exemplo, um processador (por exemplo, a unidade de processamento central (CPU), processador de sinal digital (DSP), etc.) ou qualquer bloco de hardware dedicado (por exemplo, qualquer hardware que lida com os dados do sensor (por exemplo, dados do sensor de rastreamento ocular, dados de sensor de movimento, etc.)) pode atualizar os parâmetros. O circuito VLTW 104 (por exemplo, GPU) pode ler (por exemplo, pode sempre ler, ler continuamente, periodicamente ler, etc.) a partir de uma seção de memória (por exemplo, memória em chip 110 e/ou memória de sistema 114) que inclui os parâmetros. Por conseguinte, o circuito VLTW 104 (por exemplo, GPU) pode ajustar em conformidade por primitivo em vez de para uma chamada de desenho inteira. Em algumas configurações, o circuito VLIW 104 pode renderizar diferentes níveis de detalhe em uma única chamada de desenho.

[0052] Os sistemas e métodos aqui descritos podem prover um ou mais benefícios. Por exemplo, o controlador de memória assíncrona 108 pode permitir o processamento mais eficiente ao renderizar múltiplos LODs. Em abordagens

conhecidas, múltiplas camadas de sobreposição podem ser renderizadas, em que cada camada tem um LOD diferente. Em seguida, as camadas de sobreposição podem ser misturadas. No entanto, esta abordagem é ineficiente, porque os mesmos dados são renderizados várias vezes. De acordo com os sistemas e métodos aqui divulgados, os diferentes níveis de detalhe podem ser renderizados em uma única passagem. Isto pode evitar a rerenderização dos mesmos dados e/ou pode evitar ter que misturar várias camadas com diferentes níveis de detalhe.

[0053] Adicionalmente ou em alternativa, os sistemas e métodos aqui divulgados podem permitir a mudança de forma assíncrona de LOD. Por exemplo, renderizar uma ou mais zonas de uma imagem pode ser baseado em rastreamento ocular. Por exemplo, o dispositivo eletrônico 102 pode renderizar várias zonas com diferentes níveis de detalhe, em que uma zona com um maior nível de detalhe é renderizada em uma área em que um usuário está olhando. O rastreamento ocular pode detectar quando um usuário muda para olhar para uma parte diferente da imagem. Os sistemas e métodos aqui descritos (por exemplo, o controlador de memória assíncrona 108) podem mudar imediatamente o LOD renderizado baseado em um rastreamento ocular. Por exemplo, o circuito VLIW 104 pode aumentar o nível de detalhe em uma área da imagem que o usuário tiver alterado o olhar para visualizar, mesmo durante o processamento de um conjunto de dados. Isto pode prover uma melhor experiência de usuário. Além disso, os sistemas e métodos aqui descritos (por exemplo, o controlador de memória assíncrona 108) pode reduzir o LOD em uma área da

imagem para a qual o olhar do usuário não é dirigido. Isto pode reduzir o processamento desnecessário.

[0054] O circuito VLIW 104 pode incluir um controlador de buffer de quadro não uniforme 112. O controlador de buffer de quadro não uniforme 112 (por exemplo, FBO de multirresolução) pode ser implementado em hardware ou em uma combinação de hardware e software. O controlador de buffer de quadro não uniforme 112 pode ser implementado no circuito VLIW 104, separadamente do circuito VLIW 104, ou pode ser implementado parcialmente no circuito VLIW 104 e parcialmente separado do circuito VLIW 104. Em algumas configurações, o controlador de buffer de quadro não uniforme 112 pode ser implementado como parte do pipeline gráfico 106. Em alternativa, o controlador de buffer de quadro não uniforme 112 pode ser implementado como parte da memória em chip 110. Em uma outra implementação, o controlador de buffer de quadro não uniforme 112 pode ser implementado no circuito VLIW 104, mas separado do pipeline gráfico 106 e da memória em chip 110.

[0055] O controlador de buffer de quadro não uniforme 112 pode ser configurado para acessar dinamicamente diferentes subconjuntos de um ou mais buffer (s) de quadro. Em algumas configurações, os diferentes subconjuntos de um buffer de quadro podem corresponder a diferentes níveis de detalhe, zonas com diferentes níveis de detalhe e/ou diferentes níveis de mapa de bits. Por exemplo, o controlador de buffer de quadro não uniforme 112 pode ser configurado para operar em diferentes subconjuntos de um buffer de quadro dinamicamente. Deve ser notado que pode haver dois ou mais subconjuntos de buffer de quadro. Em algumas configurações,

diferentes subconjuntos do buffer de quadro não uniforme podem corresponder a diferentes visualizações (por exemplo, uma visualização de olho esquerdo e uma visualização de olho direito para renderização estereoscópica). O buffer de quadro não uniforme pode melhorar a utilização da largura de banda. Com um buffer de quadro não uniforme, por exemplo, a visualização do olho esquerdo e a visualização do olho direito podem ser processadas simultaneamente (por exemplo, ao mesmo tempo) e salvas na memória de sistema 114 em uma passagem. Adicionalmente ou em alternativa, por exemplo, o buffer de quadro não uniforme pode ativar e/ou melhorar a utilização da memória para renderizar múltiplos LODs em uma só passagem.

[0056] Em algumas configurações, o acesso dinamicamente a diferentes subconjuntos do buffer de quadro pode incluir o acesso a pelo menos um pixel de uma primeira zona a uma taxa diferente de pelo menos um pixel de uma segunda zona. A primeira zona e a segunda zona podem corresponder a diferentes níveis de detalhe. Por exemplo, o controlador de buffer de quadro não uniforme 112 pode permitir que pixels de uma zona (com uma LOD superior, por exemplo) sejam processados em uma taxa maior do que pixels de uma outra zona (com um LOD inferior, por exemplo). Esta abordagem pode permitir economias de energia ao diminuir o processamento de uma ou mais zonas (por exemplo, zonas de menor LOD). Note-se que pode haver pixels de duas ou mais zonas que podem ser acessados em duas ou mais taxas diferentes.

[0057] Além disso ou alternativamente, o acesso de forma dinâmica a diferentes subconjuntos do buffer de quadro

pode incluir permitir troca de sub-rotina de shader. Por exemplo, troca de sub-rotina de shader pode ser realizada quando o processamento de um primeiro pixel está em andamento e processo de um segundo pixel é concluído. Por exemplo, um primeiro subconjunto do buffer de quadro pode ser alocado a um ou mais pixels processados com uma primeira sub-rotina de shader, enquanto que um segundo subconjunto do buffer de quadro pode ser alocado para um ou mais pixels processados com uma segunda sub-rotina de shader. Quando o processamento de um ou mais pixels de segunda sub-rotina de shader é concluído, o dispositivo eletrônico 102 pode trocar de sub-rotinas de shader para ocupar o segundo subconjunto do buffer de quadro. Isto pode prover um renderização mais eficiente permitindo que a capacidade do buffer de quadro seja utilizada de forma mais eficiente. Deve ser notado que dois ou mais subconjuntos do buffer de quadro podem ser utilizados para permitir troca de sub-rotina para um ou mais dos subconjuntos.

[0058] O controlador de buffer de quadro não uniforme 112 pode ser configurado para substituir uma primeira amostra processada de um primeiro pixel para uma segunda amostra de um segundo pixel para evitar o processamento da segunda amostra do segundo pixel. O primeiro pixel pode ter um número diferente de amostras do que o segundo pixel. Por exemplo, um primeiro pixel a partir de uma primeira zona pode ter um nível de detalhe maior do que um segundo pixel vizinho de uma segunda zona com um LOD inferior. O controlador de buffer de quadro não uniforme 112 pode substituir a amostra (não processada) do segundo pixel com a amostra (processada) do primeiro pixel. Deste modo, o

controlador de buffer de quadro não uniforme 112 pode reduzir o processamento. Deve ser notado que o controlador de buffer de quadro não uniforme 112 pode substituir uma ou mais amostras (não processadas) de um ou mais pixels com uma ou mais amostras (processadas) de um ou mais outros pixels.

[0059] Em algumas configurações, o dispositivo eletrônico pode incluir 102 e/ou pode ser acoplado a um ou mais (por exemplo, displays, displays head-mounted). O display (s) pode ser configurado para apresentar os diferentes níveis de detalhe.

[0060] Figura 2 é um diagrama de fluxo que ilustra uma configuração de um método 200 para renderizar múltiplos níveis de detalhe. O método 200 pode ser realizado pelo dispositivo eletrônico 102 descrito em ligação com a Figura 1.

[0061] O dispositivo eletrônico 102 pode obter 202 dados de geometria. Por exemplo, o dispositivo eletrônico 102 pode carregar dados de geometria provenientes do armazenamento (por exemplo, armazenamento em disco rígido, armazenamento removível, tal como um disco rígido externo, unidade de disco de vídeo digital (DVD), unidade de Blu-Ray, disco compacto (CD), thumb drive, etc.) para a memória (por exemplo, a memória de sistema 114 e/ou de memória em chip 110). Adicionalmente ou em alternativa, o dispositivo eletrônico 102 pode receber dados de geometria a partir de outro dispositivo. Por exemplo, o dispositivo eletrônico 102 pode receber dados de geometria através de uma interface de comunicação (por exemplo, Ethernet, Institute of Eletrônica and Electronic Engineers (IEEE) interface 802.11, interface de rede celular, etc.) e carregar os dados de geometria na

memória (por exemplo, memória de sistema 114 e/ou memória em chip 110). Exemplos de dados de geometria podem incluir pontos, linhas e/ou polígonos (por exemplo, triângulos), etc.

[0062] O dispositivo eletrônico 102 pode renderizar 204 diferentes níveis de detalhe de forma assíncrona. Isto pode ser conseguido tal como descrito acima em ligação com a Figura 1. Por exemplo, o dispositivo eletrônico 102 pode acessar a memória de forma assíncrona (por exemplo, a memória de sistema 114 e/ou memória em chip 110) para renderizar diferentes níveis de detalhe. Por exemplo, o dispositivo eletrônico 102 pode alterar um LOD para um conjunto de dados durante o processamento (em que o conjunto de dados inicialmente tinha um LOD diferente, por exemplo). Em algumas configurações, renderizar 204 diferentes níveis de detalhe de forma assíncrona pode incluir a realização de tecelagem de LOD dinâmico.

[0063] O dispositivo eletrônico 102 pode acessar dinamicamente 206 diferentes subconjuntos de um buffer de quadro. Isto pode ser conseguido tal como descrito acima em ligação com a Figura 1. Por exemplo, o dispositivo eletrônico 102 pode acessar pelo menos um pixel de uma primeira zona a uma taxa diferente de pelo menos um pixel de uma segunda zona. Adicionalmente ou em alternativa, o dispositivo eletrônico 102 pode permitir troca de sub-rotina de shader, quando o processamento de um primeiro pixel está em andamento e processo de um segundo pixel é concluído. Adicionalmente ou em alternativa, o dispositivo eletrônico 102 pode substituir uma primeira amostra processada de um primeiro

pixel para uma segunda amostra de um segundo pixel para evitar o processamento da segunda amostra do segundo pixel.

[0064] A figura 3 é um diagrama de blocos que ilustra uma configuração de um dispositivo eletrônico 302 no qual os sistemas e métodos para renderizar múltiplos níveis de detalhe podem ser implementados. O dispositivo eletrônico 302 descrito em ligação com a Figura 3 pode ser um exemplo do dispositivo eletrônico 102 descrito em conexão com a Figura 1. Em algumas configurações, um ou mais dos componentes ou elementos do dispositivo eletrônico 302 podem ser semelhantes aos componentes ou elementos correspondentes descritos em ligação com a Figura 1. Um ou mais dos componentes ou elementos podem ser implementados em hardware (por exemplo, circuitos) ou uma combinação de hardware e de software (por exemplo, um processador com instruções).

[0065] Em algumas configurações, o dispositivo eletrônico 302 pode incluir um circuito de palavra de instrução muito longa (VLIW) 304, memória de sistema 314, um ou mais displays 332, um ou mais sensores de imagem 316, um ou mais sistemas ópticos 318, uma ou mais interfaces de comunicação 320, uma ou mais antenas 322 e/ou um ou mais processadores 324. Dois ou mais dos elementos ou componentes descritos em ligação com a Figura 3 podem estar em comunicação eletrônica um com o outro. Por exemplo, o circuito VLIW 304 pode ser acoplado a (por exemplo, em comunicação eletrônica com) a memória de sistema 314. O circuito VLIW 304 pode ser um dispositivo de renderização de imagens, tal como descrito acima em ligação com uma ou mais das Figuras 1 e 2. Em algumas configurações, a imagem (s) resultante pode ser apresentada em um ou mais displays 332.

O circuito VLIW 304 pode ser separado do processador 324, em algumas implementações. Em outras implementações, o circuito VLIW 304 pode ser integrado no processador 324.

[0066] A interface de comunicação 320 pode permitir que o dispositivo eletrônico 302 se comunique com um ou mais outros dispositivos eletrônicos. Por exemplo, a interface de comunicação 320 pode prover uma interface para comunicações com e/ou sem fio. Em algumas configurações, a interface de comunicação 320 pode ser acoplada a uma ou mais antenas 322 para a transmissão e/ou recepção de sinais de radiofrequência (RF). Adicionalmente ou em alternativa, a interface de comunicação 320 pode permitir um ou mais tipos de comunicação fixa (por exemplo, Barramento Universal Serial (USB), Ethernet, Arranjo de Gráficos de Vídeo (VGA), Interface Digital Visual (DVI), Interface Multimídia de Alta Definição (HDMI), Porta de Display, link de Alta Definição móvel (MHL), etc.).

[0067] Em algumas configurações, várias interfaces de comunicações 320 podem ser implementada e/ou utilizadas. Por exemplo, uma interface de comunicação 320 pode ser uma interface de comunicação celular 320 (por exemplo, 3G, Evolução de Longo Prazo (LTE), CDMA, etc.), uma outra interface de comunicação 320 pode ser uma interface Ethernet, uma outra interface de comunicação 320 pode ser uma interface de barramento serial universal (USB), ainda outra interface de comunicação 320 pode ser uma interface de rede de área local sem fio (WLAN) (por exemplo, Interface 802.11 do Institute of Eletrônica and Electronics Engineers (IEEE)) e ainda outra interface de comunicação 320 pode ser uma interface Bluetooth.

[0068] Em algumas configurações, o dispositivo eletrônico 302 pode executar uma ou mais das funções, processos, métodos, etapas, etc., descritos em ligação com uma ou mais das Figuras 1, 3-5, 9, 11, 13 e 15-24. Adicionalmente ou em alternativa, o dispositivo eletrônico 302 pode incluir uma ou mais das estruturas descritas em ligação com uma ou mais das Figuras 1, 3-5, 9, 11, 13 e 15-24.

[0069] A memória de sistema 314 pode armazenar instruções e/ou dados. A memória de sistema 314 descrita em ligação com a Figura 3 pode ser um exemplo do memória de sistema 114 descrito em ligação com a Figura 1. O circuito VLIW 304 pode acessar (por exemplo, ler a partir de e/ou gravar na) a memória de sistema 314. A memória de sistema 314 pode também ser acessível pelo processador (s) 324. A memória de sistema 314 pode armazenar dados de geometria 328. Em algumas configurações, a memória de sistema 314 pode incluir um buffer de quadro 330a. O buffer de quadro 330a pode ser uma parte da memória de sistema 314 para armazenar os dados de imagem. Os dados de imagem armazenados em buffer podem ser apresentados, por um ou mais displays 332, em algumas configurações.

[0070] O circuito VLIW 304 pode incluir um ou mais elementos de renderização de imagens. Alguns exemplos dos elementos pode incluir um front end (EF), processador de vértice (VPs), coletor de vértice (VF), shader de vértice (VS), shader de controle de tecelagem (TCS), hull shader (HS), tessellator (por exemplo, máquina de tecelagem) (TESS), shader de avaliação de tecelagem (TES), shader de domínio (DS), shader de shader (GS), rasterizador (RAST),

shader de fragmento (FS), pixel shader (PS), mosaico, unidade de saída de renderização (ROP), clipper (clip) e/ou saída de fluxo (SOL), etc. Em algumas configurações, um ou mais destes elementos podem ser referidos como um pipeline gráfico.

[0071] O circuito VLIW 304 pode incluir memória 310. A memória em chip 310 descrita em ligação com a Figura 3 pode ser um exemplo da memória em chip 110 descrita em conexão com a Figura 1. Em algumas configurações, a memória em chip 310 pode incluir um buffer de quadro 330b (além de ou em alternativa ao buffer de quadro 330a na memória de sistema 314, por exemplo). O buffer de quadro pode ser uma parte da memória em chip 310 para armazenar dados de imagem. Os dados de imagem armazenados em buffer podem ser apresentados, por um ou mais displays 332, em algumas configurações.

[0072] Em algumas configurações, o circuito VLIW 304 pode incluir um controlador de memória assíncrona 308. O controlador de memória assíncrona 308 pode ser implementado em hardware ou em uma combinação de hardware e software. O controlador de memória assíncrona 308 pode ser implementado no circuito VLIW 304, separadamente do circuito VLIW, ou pode ser implementado parcialmente no circuito VLIW 304 e parcialmente separado do circuito VLIW 304. O controlador de memória assíncrona 308 pode ser configurado para acessar a memória (por exemplo, memória em chip 310 e/ou memória de sistema 314) de forma assíncrona. Deve ser notado que o controlador de memória assíncrona 308 pode utilizar (por exemplo, acesso) a memória em chip 110 e/ou a memória de sistema 114 (por exemplo, um ou ambos). Algumas configurações podem depender de um caso de uso. Por exemplo, se os dados

do sensor (por exemplo, dados de sensor de imagem 316) estão sendo gravados a uma taxa elevada (por exemplo, 1000 hertz (Hz)), em seguida, o cache pode perder tempo, bateria e/ou outros recursos. O controlador de memória assíncrona 308 descrito em ligação com a Figura 3 pode ser um exemplo do controlador de memória assíncrona 108 descrito em conexão com a Figura 1. O controlador de memória assíncrona 308 descrito em ligação com a Figura 3 pode operar como descrito em ligação com o controlador de memória assíncrona 108 descrito em ligação com a Figura 1.

[0073] O circuito VLIW 304 pode incluir um controlador de buffer de quadro não uniforme 312. O controlador de buffer de quadro não uniforme 312 pode ser implementado em hardware ou uma combinação de hardware e software. O controlador de buffer de quadro padrão não uniforme 312 pode ser implementado no circuito VLIW 304, separadamente do circuito VLIW 304, ou pode ser implementado parcialmente no circuito VLIW 304 e parcialmente separado do circuito VLIW 304. Em algumas configurações, o controlador de buffer de quadro não uniforme 312 pode ser implementado como parte do pipeline gráfico. Em alternativa, o controlador de buffer de quadro não uniforme 312 pode ser implementado como parte da memória em chip 310. Em outra implementação, o controlador de buffer de quadro não uniforme 312 pode ser implementado no circuito VLIW 304, mas separado do pipeline gráfico 306 e da memória em chip 310. O controlador de buffer de quadro não uniforme 312 descrito em ligação com a Figura 3 pode ser um exemplo do controlador de buffer de quadro não uniforme 112 descrito em conexão com a Figura 1. O controlador de buffer de quadro não uniforme 312 descrito em

ligação com a Figura 3 pode operar como descrito em ligação com o controlador de buffer de quadro não uniforme 112 descrito em ligação com a Figura 1.

[0074] Em algumas configurações, o dispositivo eletrônico 302 pode realizar renderização de passagem única multi-LOD. Renderização multi-LOD pode incluir renderização de diferentes zonas de uma imagem com diferentes níveis de detalhe. Renderização de passagem única multi-LOD pode incluir renderizar zonas diferentes com diferentes níveis de detalhe (por exemplo, níveis de tecelados de detalhe) em uma única passagem (sem processar repetidamente todo o pixel, por exemplo). Renderização de passagem única foveada pode ser um exemplo de renderização de passagem única multi-LOD, em que diferentes zonas com diferentes níveis de detalhe podem ser baseadas em rastreamento ocular. Por exemplo, uma zona pode ser uma área circular (ou uma região de outra forma) aproximadamente centrada no ponto de uma visualização (por exemplo, imagem) onde o usuário está olhando. Adicionalmente ou em alternativa, renderização de passagem única (por exemplo, renderização de passagem única multi-LOD) pode incluir a realização de uma passagem (por exemplo, passagem de processamento) sobre os dados e/ou textura para ser renderizada, mesmo que ambas uma visualização de olho esquerdo e uma visualização de olho direito precise de ser processada. Por conseguinte, algumas configurações dos sistemas e métodos aqui descritos podem utilizar o máximo dos dados para ambos os pontos de visualização do olho, tanto quanto possível, antes de mover os dados entre a memória de sistema e a memória em chip (por exemplo, a memória da GPU). Em algumas configurações, o controlador de memória

assíncrona 308 e/ou o controlador de buffer de quadro não uniforme 312 pode permitir a realização de uma renderização de única passagem multi-LOD (por exemplo, renderização foveada de passagem única). Por exemplo, o controlador de memória assíncrona 308 pode permitir parâmetro de leitura (s) (por exemplo, o parâmetro nível de detalhe (LOD)) a partir de um bloco de memória volátil (por exemplo, adaptar dinamicamente durante uma chamada de desenho). Por exemplo, o controlador de memória assíncrona 308 pode permitir um ou mais de um shader de controle de tecelagem (TCS), tessellator (TESS), shader de avaliação de tecelagem (TES), shader de geometria (GS), rasterizador e shader de fragmento (FS) para acessar um ou mais parâmetros a partir de um bloco de memória volátil (por exemplo, para se adaptar dinamicamente durante uma chamada de desenho).

[0075] Em algumas configurações, o circuito VLIW 304 pode incluir um rasterizador. O rasterizador pode permitir MSAA para executar mais amostragem dos primitivos, os quais por sua vez podem ativar mais avaliações de shader de fragmento (FS). Além disso, amostragem dos primitivos pode ser realizada para evitar artefatos como jaggies, que são devido a amostras insuficientes sendo processadas e medidas.

[0076] O dispositivo eletrônico 302 pode obter uma ou mais imagens (por exemplo, imagens digitais, quadros de imagens, vídeo, etc.). As uma ou mais imagens podem ser imagens de um ou mais usuários e/ou uma cena (por exemplo, um ou mais objetos e/ou fundo). Por exemplo, o dispositivo eletrônico 302 pode incluir um ou mais sensores de imagem 316 e/ou um ou mais sistemas ópticos 318 (por exemplo,

lentes) que se concentram as imagens de objetos que se encontram dentro do campo de visualização do sistema óptico (s) 318 para o sensor de imagem (s) 316. Em algumas configurações, o sensor (s) de imagem 316 pode captar uma ou mais imagens. O sensor (s) de imagem 316 e/ou sistema óptico (s) 318 pode ser acoplado a e/ou controlado pelo processador 324.

[0077] Adicionalmente ou em alternativa, o dispositivo eletrônico 302 pode solicitar e/ou receber uma ou mais imagens a partir de outro dispositivo (por exemplo, um sensor de imagem externo acoplado ao dispositivo eletrônico 302, um servidor de rede, câmera de tráfego, câmera drop, câmera de automóvel, câmera web, etc.). Em algumas configurações, o dispositivo eletrônico 302 pode solicitar e/ou receber uma ou mais imagens através da interface de comunicação 320. Por exemplo, o dispositivo eletrônico 302 pode ou não incluir uma câmara (por exemplo, um sensor de imagem 316 e/ou sistema óptico 318) e pode receber imagens de um dispositivo remoto.

[0078] Em algumas configurações, os um ou mais sensores de imagem 316 incluídos no dispositivo eletrônico 302 e/ou um ou mais outros sensores de imagem acoplados ao dispositivo eletrônico 302 podem ser orientados para captar uma ou mais imagens de um usuário. Por exemplo, um smartphone pode incluir uma câmara montada na mesma face de um display (por exemplo, display 332). Desta forma, a câmara pode capturar imagens de um usuário enquanto o usuário está olhando para o display. Em um outro exemplo, uma câmara pode ser montada em um console e/ou volante de um veículo para capturar imagens de um usuário (por exemplo, condutor).

[0079] Adicionalmente ou em alternativa, os um ou mais sensores de imagem 316 incluídos no dispositivo eletrônico 302 e/ou um ou mais outros sensores de imagem acoplados ao dispositivo eletrônico 302 podem ser orientados para capturar cenário. Por exemplo, uma ou mais câmeras podem ser montadas na parte de trás de um smartphone. Em um outro exemplo, uma ou mais câmaras podem ser montadas no exterior de um veículo. Em algumas configurações, o dispositivo eletrônico 302 pode utilizar as imagens da paisagem em aplicações de realidade aumentada (AR), aplicações de realidade virtual (VR) e/ou outras aplicações (por exemplo, visualização surround, assistência ao condutor, sistemas de assistência ao condutor avançados (ADAS), etc.).

[0080] Em algumas configurações, a imagem (s) capturada pelo sensor (s) de imagem 316 e/ou recebida através da interface de comunicação (s) 320 pode ser utilizada pelo dispositivo eletrônico 302 para o rastreamento ocular. Por exemplo, o processador 324 pode incluir e/ou implementar um rastreamento ocular 326. O rastreamento ocular 326 pode controlar o olhar (por exemplo, direção do olhar) de um ou mais usuários. Por exemplo, o rastreamento ocular 326 pode localizar o olho (s) de um ou mais usuários na imagem (s) e pode estimar o olhar (por exemplo, direção do olhar) do usuário (es). Por exemplo, o rastreamento ocular 326 pode determinar (por exemplo, estimar) onde um usuário está olhando em um display (por exemplo, display 332 ou um display externo) e/ou quando um usuário está olhando de uma imagem apresentada por um display. Em outras palavras, o rastreamento ocular 326 pode determinar (por exemplo,

estimativa) para onde o olhar de um usuário está direcionado em relação a uma imagem e/ou display.

[0081] Os sistemas e métodos aqui divulgados podem permitir a mudança de forma assíncrona de LOD e/ou renderização de múltiplos níveis de detalhe, em uma única passagem. Em algumas configurações, renderizar uma ou mais zonas de uma imagem pode ser baseado em rastreamento ocular. As zonas da imagem podem ter diferentes níveis de detalhe. Por exemplo, o dispositivo eletrônico 302 pode processar várias zonas com diferentes níveis de detalhe, em que uma zona com um maior nível de detalhe é renderizada em uma área em que um usuário está atualmente olhando. Em algumas configurações, isso pode ser referido como renderização foveada. O rastreamento ocular 326 pode produzir informações de rastreamento ocular. As informações de rastreamento ocular podem indicar para onde o olhar de um usuário está direcionado (por exemplo, olhares dos usuários são direcionados) em relação a um display e/ou imagem. Em algumas configurações, o processador 324 e/ou o circuito VLIW 304 pode determinar um ou mais parâmetros de nível de detalhe com base na informação de rastreamento ocular. O parâmetro (s) de LOD pode indicar um LOD a ser renderizado por uma ou mais zonas de uma imagem. O parâmetro (s) de LOD pode ser armazenado na memória de sistema 314 310 e/ou memória em chip. O parâmetro de LOD (s) pode ser atualizado de forma assíncrona em relação a um conjunto de dados (por exemplo, chamada de desenho) que está sendo processado pelo circuito VLIW 304. O controlador de memória assíncrona 308 pode ler o parâmetro de LOD (s) (a partir da memória de sistema 314 e/ou memória em chip 310) durante o processamento do conjunto

de dados (por exemplo, chamada de desenho). Por exemplo, o controlador de memória assíncrona 308 pode atualizar definições de tecelagem durante o processamento do conjunto de dados (por exemplo, chamada de desenho).

[0082] O rastreamento ocular 326 pode detectar quando um usuário muda a ótica para olhar para uma parte diferente da imagem. Os sistemas e métodos aqui descritos (por exemplo, o controlador de memória assíncrona 308) pode mudar imediatamente o LOD renderizado baseado no rastreamento ocular (por exemplo, a informação de rastreamento ocular e/ou parâmetro (s) de LOD). Por exemplo, o controlador de memória assíncrona 308 pode ler um ou mais parâmetros (por exemplo, parâmetros de LOD), que podem mudar durante o processamento de um conjunto de dados (por exemplo, chamada de desenho). Por exemplo, o circuito VLIW 304 pode aumentar o nível de detalhe em uma área (por exemplo, zona) da imagem que o usuário alterou a ótica para visualizar, mesmo durante o processamento de um conjunto de dados. Adicionalmente ou em alternativa, os sistemas e métodos aqui descritos (por exemplo, o controlador de memória assíncrona 308) pode reduzir o LOD em uma área (por exemplo, zona) da imagem para a qual o olhar do usuário já não é dirigido. Em algumas configurações, a zona pode ser uma área circular (ou a uma região de outro formato) aproximadamente centrada no ponto de visualização (por exemplo, imagem) onde o usuário está olhando. Em outras configurações, uma zona pode ser toda a área de uma imagem que corresponde a uma janela que o usuário está olhando.

[0083] Em algumas configurações, o dispositivo eletrônico 302 pode incluir um aplicativo de câmera e/ou um

display 332. Quando a aplicativo de câmara está em execução, as imagens de objetos que se encontram dentro do campo de visualização do sistema óptico 318 podem ser capturadas pelo sensor de imagem 316. As imagens que estão sendo capturadas pelo sensor de imagem 316 podem ser apresentadas no display 332. Em algumas configurações, tais imagens podem ser apresentadas em sucessão rápida a uma taxa de quadro relativamente elevada de modo que, em qualquer dado momento em tempo, os objetos que se encontram dentro do campo de visualização do sistema óptico 318 são apresentados no display 332.

[0084] Deve ser notado que um ou mais dos elementos ilustrados no dispositivo eletrônico 302 podem ser opcionais. Por exemplo, um ou mais dos elementos ilustrados no dispositivo eletrônico 302 podem ou não ser incluídos e/ou implementados. Adicionalmente ou em alternativa, um ou mais dos elementos ilustrados no processador 324 e/ou circuito VLIW 304 podem ser implementados separadamente a partir do processador 324 e/ou circuito VLIW 304 (por exemplo, em outros circuitos, sobre outro processador, sobre um dispositivo eletrônico separado, etc.). Por exemplo, o sensor de imagem 316, o sistema óptico 318, a interface de comunicação 320, o processador 324, o rastreamento ocular 326 e/ou o display 332 não podem ser implementados no dispositivo eletrônico 302. Em algumas configurações, o dispositivo eletrônico 302 pode receber imagens e/ou informações de rastreamento ocular de outro dispositivo. Adicionalmente ou em alternativa, um ou mais displays podem ser implementados em um ou mais dispositivos separados.

[0085] Em algumas configurações, o dispositivo eletrônico 302 pode apresentar uma interface de usuário 334 no display 332. Por exemplo, a interface de usuário 334 pode permitir que um usuário interaja com o dispositivo eletrônico 302. Em algumas configurações, o display 332 pode ser uma tela de toque que recebe a entrada a partir de toque físico (por um dedo, caneta ou outra ferramenta, por exemplo). Adicionalmente ou em alternativa, o dispositivo eletrônico 302 pode incluir ou ser associado a uma outra interface de entrada. Por exemplo, o dispositivo eletrônico 302 pode incluir uma câmara de frente para um usuário e pode detectar gestos de usuário (por exemplo, gestos de mão, gestos de braço, rastreamento ocular, piscada dos olhos, etc.). Em outro exemplo, o dispositivo eletrônico 302 pode ser acoplado a um mouse e pode detectar um clique do mouse. Deve ser notado que nenhuma entrada do usuário pode ser necessária em algumas configurações.

[0086] Em algumas configurações, o dispositivo eletrônico 302 pode opcionalmente incluir um sensor de movimento (por exemplo, giroscópio (s), acelerômetro (s), etc.). Por exemplo, o sensor de movimento pode detectar movimento e/ou uma orientação (do dispositivo eletrônico 302, por exemplo). Em algumas configurações, a imagem sendo processada pode ser baseada em informações de movimento e/ou orientação. Por exemplo, o dispositivo eletrônico 302 pode ser um display montado na cabeça (por exemplo, um fone de ouvido de realidade virtual (RV), um fone de ouvido de realidade aumentada (AR), etc.), um veículo (por exemplo, automóvel, drone, avião, etc.), um dispositivo móvel (por exemplo, smartphone) ou outro dispositivo que pode utilizar

informações de movimento e/ou orientação para processar uma imagem. Por exemplo, a informação de movimento e/ou orientação pode indicar uma direção de visualização geral em uma cena renderizada. O dispositivo eletrônico 302 pode processar a imagem com base nesta direção de observação, em complemento ou em alternativa, a partir da informação de rastreamento ocular em algumas configurações. O uso de um sensor de movimento pode ilustrar um outro aspecto da operação assíncrona do controlador de memória assíncrona 308. Por exemplo, um sensor de movimento pode prover a informação do movimento e/ou informação de orientação para a memória (por exemplo, memória de sistema 314 e/ou memória em chip 310) a uma taxa particular (por exemplo, 1000 hertz (Hz)), enquanto que o circuito VLIW 304 pode renderizar quadros a uma taxa relativamente mais baixa (por exemplo, 30 quadros por segundo (fps)). O controlador de memória assíncrona 308 pode de forma assíncrona alterar uma ou mais operações de renderização (por exemplo, tecelagem) no meio de renderização de um conjunto de dados com base na informação do movimento e/ou informação de orientação.

[0087] A Figura 4 é um diagrama que ilustra um exemplo de um buffer de quadro não uniforme 430. Neste exemplo, o buffer de quadro 430 tem uma capacidade de  $2n \times 2m$  pixels. Em algumas configurações, o dispositivo eletrônico 302 pode, opcionalmente, ladrilhar 432 o buffer de quadro 430 em mosaicos 436. No exemplo ilustrado na Figura 4, quatro mosaicos 436 de tamanho  $(n, m)$  estão ilustrados. Em algumas modalidades, memória em chip 310 ou uma parte da memória em chip 310 pode ser limitada ao tamanho de um mosaico 436. Deste modo, o dispositivo eletrônico 302 pode

dividir o buffer de quadro 430 em mosaicos 436. Em outras implementações, o memória em chip 310 ou uma parte da memória em chip 310 pode ser grande o suficiente para acomodar todo o buffer de quadro 430 e nenhuma ladrilhagem 434 pode não ser realizada.

[0088] O dispositivo eletrônico 302 pode subdividir o buffer de quadro 438 430 e/ou mosaicos 436 em porções dimensionadas 440. Por exemplo, o buffer de quadro 430 e/ou mosaicos 436 pode ser subdivididos para se encaixarem em várias resoluções e/ou níveis de detalhe. No exemplo ilustrado na Figura 4, as porções dimensionadas 440 são de tamanho (n/c, m/c), em que c é um fator de escala. O fator de escala c pode variar e/ou pode ser selecionado com base na configuração.

[0089] O dispositivo eletrônico 302 (por exemplo, controlador de buffer de quadro não uniforme 312) pode acessar dinamicamente diferentes subconjuntos 442a-c do buffer de quadro 430. No exemplo, ilustrado na Figura 4, o dispositivo eletrônico 302 pode acessar dinamicamente o subconjunto de buffer A 442a, subconjunto de buffer B 442b e 442c subconjunto de buffer C. Deve ser notado que menos ou mais subconjuntos de buffer podem ser utilizados. Deve também notar-se que a totalidade ou apenas uma parte do buffer de quadro 430 pode ser ocupada por subconjuntos de buffer. Por exemplo, as outras porções não hachuradas do buffer de quadro podem ser utilizadas por um ou mais de outros subconjuntos.

[0090] Acesso dinâmico (por exemplo, a leitura e/ou gravação) dos diferentes subconjuntos de buffer de quadro (por exemplo, os subconjuntos de buffer 442a-c) pode permitir uma ou mais abordagens para prover um maior controle

e/ou eficiência em imagens de renderização. Em algumas abordagens, acesso dinâmico de diferentes subconjuntos de buffer de quadro pode permitir renderização de passagem única multi-LOD. Isso ocorre porque renderização de uma zona em um LOD não precisa ser concluída antes de renderização de outra zona em um LOD diferente poder ser iniciada. Nesta abordagem, um ou mais dos subconjuntos de buffer 442a-c podem corresponder a diferentes zonas da imagem. Por exemplo, subconjunto de buffer A 442a pode corresponder a uma zona com um LOD maior, subconjunto de buffer B 442b pode corresponder a uma zona com um LOD médio e subconjunto de buffer C 442c pode corresponder a uma zona com um LOD inferior. Por exemplo, o circuito VLIW 304 pode renderizar uma maior zona de LOD, que pode ser gravado no subconjunto de buffer A 442a, enquanto ao mesmo tempo ou sequencialmente renderiza uma zona de LOD médio, que pode ser gravado no subconjunto de buffer B 442b. Deve ser notado que as zonas de LOD diferentes podem ser processadas em qualquer ordem (por exemplo, a partir do LOD de mais alto para o mais baixo, na ordem inversa ou outra ordem). Adicionalmente ou em alternativa, acesso dinâmico de diferentes subconjuntos de buffer de quadro pode permitir o processamento de diferentes zonas em diferentes taxas (por exemplo, um subconjunto de buffer de quadro maior pode ser alocado para uma zona a ser processada mais rapidamente, enquanto um subconjunto de buffer de quadro menor pode ser alocado para outra zona a ser processada mais devagar).

[0091] Em algumas abordagens, acesso dinâmico de diferentes subconjuntos de buffer de quadro pode permitir processos de shader paralelos (por exemplo, sub-rotinas).

Por exemplo, processos de shader diferentes podem ser executados em paralelo, em que um processo de shader de acessa um subconjunto do buffer de quadro e um outro processo de shader acessa outro subconjunto do buffer de quadro. Adicionalmente ou em alternativa, o acesso dinâmico de diferentes subconjuntos de buffer de quadro pode permitir troca de sub-rotina de shader. Por exemplo, uma primeira sub-rotina de shader pode ser executada a qual acessa subconjunto de buffer A 442a. Enquanto a primeira sub-rotina de shader está em andamento, uma segunda sub-rotina de shader que acessa subconjunto de buffer B 442b pode terminar. A segunda sub-rotina de shader pode ser trocada por outra sub-rotina de shader (que pode acessar o subconjunto de buffer B 442b, por exemplo), enquanto a primeira sub-rotina de shader está em andamento. Isso pode permitir uma maior eficiência na utilização do espaço do buffer de quadro, em vez de ter que esperar que uma sub-rotina de shader seja concluída antes que outra possa ser iniciada.

[0092] Em algumas abordagens, acesso dinâmico de diferentes subconjuntos de buffer de quadro pode permitir a substituição de amostra. Por exemplo, o controlador de buffer de quadro não uniforme 312 pode trocar uma ou mais amostras de um pixel processados para uma ou mais amostras de outro pixel. Em algumas configurações, o circuito VLIW 304 pode executar antialiasing (por exemplo, antialiasing multiamostra (MSAA)). No desempenho antialiasing, pixels diferentes podem ter diferentes números de amostras. Por exemplo, um primeiro pixel (por exemplo, "fino") no subconjunto de buffer A 442a pode ter 4 amostras e pode utilizar shader de maior qualidade, enquanto um próximo

segundo (por exemplo, vizinho) pixel (por exemplo, "grosseiro") no subconjunto de buffer B 442b pode ter uma amostra e usar shader de qualidade inferior. Em vez de executar o processamento (por exemplo, o shader) sobre o segundo pixel, o controlador de buffer de quadro não uniforme 312 pode substituir uma amostra do primeiro pixel por uma amostra do segundo pixel. Isto pode evitar a realização de qualquer um dos processamentos para o segundo pixel. Adicionalmente ou em alternativa, o controlador de buffer de quadro não uniforme 312 pode executar substituição de amostra ao processar mapa de bit. Por exemplo, o controlador de buffer de quadro não uniforme 312 pode substituir uma amostra a partir de um nível mais elevado (em subconjunto de buffer A 442a, por exemplo) para uma amostra a partir de um nível inferior (em subconjunto de buffer B 442b, por exemplo). Isto pode evitar alguns ou processar em um ou mais níveis inferiores.

[0093] O controlador de buffer de quadro não uniforme 312 pode permitir o envio dinâmico de processamento de shader (por exemplo, sub-rotinas, núcleos, etc.), pode permitir a substituição do pixel de amostra e/ou pode prover um buffer de quadro multi-LOD. Tal como ilustrado na Figura 4, o buffer de quadro 430 pode armazenar dados de imagem que podem ser transmitidos como múltiplos alvos renderizados 444.

[0094] Listagem (1) ilustra um exemplo de pseudocódigo que ilustra uma abordagem para a funcionalidade de buffer de quadro não uniforme. Deve ser notado que GLSL é uma abreviação para Linguagem de Shading OpenGL e SSBO é uma abreviação para Shader Storage Buffer Object. Listagem

(1) ilustra a forma como várias zonas com diferentes níveis de detalhe

```

    evitar main() { //computar buffer_subset_A usando
    GLSL + texturas arbitrárias.

        //imageStore()/buffer de cor/SSBO como mecanismo
        de gravação.

    }

    evitar         main_lod_1(samplerTile/imageTile
    red_square) {
        //computar quadrado verde usando GLSL + texturas
        arbitrárias

        //e     possivelmente      textura(red_square,
        uv)/imageRead()

    }

    ...

    evitar main_lod_n(samplerTile last_square) {
        //"_lod_<n>()"     seria     acionado     após     a
        renderização de nível 0 (buffer_subset_A).

        //poderia ser simulado via um triângulo de tela
        cheia e alterando porta de visualização.

        //"_lod_<i>()" pode quer executar direto [0 a n
        - 1] ou reverso [n - 1 a 0].
    }

```

Listagem (1)

[0095] A Figura 5 é um diagrama de fluxo que ilustra uma configuração mais específica de um método 500 para renderizar múltiplos níveis de detalhe. O método 500 pode ser realizado pelo dispositivo eletrônico 302 descrito em ligação com a Figura 3.

[0096] O dispositivo eletrônico 302 pode obter 502 dados geométricos (por exemplo, dados de geometria 328). Isto pode ser conseguido como descrito em ligação com uma ou mais das Figuras 1-3. Por exemplo, o dispositivo eletrônico 302 pode carregar dados de geometria do armazenamento para a memória (por exemplo, a memória de sistema 314 e/ou memória em chip 310) e/ou pode receber dados de geometria a partir de outro dispositivo e carregar os dados de geometria para a memória (por exemplo, memória de sistema 314 e/ou memória em chip 310).

[0097] O dispositivo eletrônico 302 pode executar 504 rastreamento ocular. Isto pode ser conseguido como descrito em ligação com a Figura 3. Por exemplo, o dispositivo eletrônico 302 pode obter (por exemplo, capturar e/ou receber) uma ou mais imagens e pode determinar para onde o olhar de um usuário está direcionado (por exemplo, onde um usuário está olhando em uma tela e/ou uma imagem que está sendo apresentada no display) com base em uma ou mais imagens.

[0098] O dispositivo eletrônico 302 pode renderizar 506 diferentes níveis de detalhe de modo assíncrono baseado em um rastreamento ocular. Isto pode ser conseguido tal como descrito acima em ligação com uma ou mais das Figuras 1-3. Por exemplo, o dispositivo eletrônico 302 pode acessar de forma assíncrona a memória (por exemplo, memória de sistema 314 e/ou memória em chip 310) para renderizar diferentes níveis de detalhe com base no rastreamento ocular. Por exemplo, o dispositivo eletrônico 302 pode alterar um LOD para um conjunto de dados durante o processamento (em que o conjunto de dados inicialmente tinha

um LOD diferente, por exemplo) quando a zona para o conjunto de dados muda com base no rastreamento ocular. Em algumas configurações, o dispositivo eletrônico 302 pode aumentar o nível de detalhe para o conjunto de dados quando o olhar do usuário é dirigido para uma área da imagem que corresponde ao conjunto de dados. Adicionalmente ou em alternativa, o dispositivo eletrônico 302 pode diminuir o LOD para o conjunto de dados quando a direção do olhar de um usuário se afasta de uma área da imagem correspondente ao conjunto de dados.

[0099] Em algumas configurações, renderizar 506 diferentes níveis de detalhe de forma assíncrona pode incluir a realização de tecelagem de LOD dinâmico. Por exemplo, a memória (por exemplo, a memória de sistema 314 e/ou memória em chip 310) pode ser atualizada para indicar uma alteração no parâmetro (s) de tecelagem de LOD (por exemplo, um grau de tecelagem). O controlador de memória assíncrona 308 pode acessar o parâmetro de tecelagem de LOD atualizado para alterar a tecelagem de LOD de forma assíncrona.

[00100] O dispositivo eletrônico 302 pode acessar dinamicamente 508 diferentes subconjuntos de um buffer de quadro. Isto pode ser conseguido tal como descrito acima em ligação com uma ou mais das Figuras 1-4. Por exemplo, o dispositivo eletrônico 302 pode acessar pelo menos um pixel de uma primeira zona a uma taxa diferente de pelo menos um pixel de uma segunda zona. Adicionalmente ou em alternativa, o dispositivo eletrônico 302 pode permitir troca de sub-rotina de shader, quando o processamento de um primeiro pixel está em andamento e o processo de um segundo pixel é concluído. Adicionalmente ou em alternativa, o dispositivo

eletrônico 302 pode substituir uma primeira amostra processada de um primeiro pixel para uma segunda amostra de um segundo pixel para evitar o processamento da segunda amostra do segundo pixel.

[00101] O dispositivo eletrônico 302 pode apresentar 510 a imagem renderizada. Por exemplo, o dispositivo eletrônico 302 pode fluir para fora dados de imagem que foi renderizada 506 com diferentes níveis de detalhe de forma assíncrona. Em algumas configurações, os dados de imagem podem ser provados a um display integrado 332. Em outras configurações, os dados de imagem podem ser provados a um display remoto que se encontra em comunicação com o dispositivo eletrônico 302. Por exemplo, o dispositivo eletrônico 302 pode prover os dados de imagem para um display que é acoplado ao dispositivo eletrônico 302 (via VGA, DVI, HDMI, etc.)

[00102] A Figura 6 é um diagrama de fluxo que ilustra um exemplo do método 600 para renderizar foveado. Neste método, os dados de geometria são obtidos 602. Rastreamento ocular também é realizado 604. Uma camada grosseira é processada 606, uma camada média é renderizada 608 e uma camada fina é renderizada 610. Por exemplo, a camada fina pode ser renderizada 610 em uma imagem, onde o olhar de um usuário é direcionado. A camada média pode ser renderizada 608 em uma área maior, que se sobrepõe à área de camada fina. A camada grossa pode ser renderizada 606 em uma área ainda maior (por exemplo, a totalidade da imagem) que inclui a área de camada média e a área de camada fina. Por conseguinte, pode haver uma sobreposição significativa entre as camadas. A camada grossa, a camada média e a camada fina

podem ser fundidas 612 com antialiasing. A imagem resultante pode ser emitida 614 para o buffer de quadro.

[00103] Conforme pode ser observado, o método pode incluir 600 renderizar quantidades significativas de áreas que se sobrepõem. Por exemplo, a área de detalhe fino pode ser renderizada três vezes: uma vez na camada grossa, uma vez na camada média, e uma vez na camada fina. Isto pode ser ineficiente uma vez que executa múltiplas renderizações passando sobre os mesmos pixels.

[00104] A Figura 7 ilustra um exemplo de renderização foveada. Em particular, a Figura 7 ilustra um exemplo de uma cena para renderização foveada. A cena para renderização foveada inclui uma área de detalhe grosso 748, uma área de detalhe médio 750 e uma área de detalhe fino 752. Tal como descrito em conexão com a Figura 6, uma abordagem para renderização foveada inclui renderizar a camada grossa 754, tornando a camada média 756 e renderizar a camada fina 758. Em seguida, a fusão e mistura 760 pode ser realizada sobre a camada grossa 754, camada média 756 e a camada fina 758 para produzir uma imagem misturada final 762. Por exemplo, máscara de mistura A 764a entre a camada média 756 e camada grossa 754 pode ser usada para misturar a camada média 756 e a camada grossa 754. Além disso, máscara de mistura B 764b entre a camada média 756 e a camada fina 758 pode ser usada para misturar a camada média 756 e a camada fina 758. Como pode ser observado, esta abordagem renderiza os mesmos pixels em passagens múltiplas, conduzindo à ineficiência. Deve ser notado que uma ou mais das máscaras de mistura 764a-b podem prover uma mistura ponderada (por exemplo, gradual), em

algumas configurações, em que o peso da mistura aumenta das bordas exteriores das máscaras de mistura para área interior. Deve também ser notado que a camada grosseira 754 e/ou a imagem misturada final 762 estão ilustradas na Figura 7 como subconjuntos de toda a cena para conveniência. No entanto, a camada grosseira 754 e/ou a imagem misturada final 762 pode cobrir toda a cena.

[00105] A figura 8 é um diagrama de blocos que ilustra um exemplo de um dispositivo eletrônico 866 incluindo um circuito de palavra de instrução muito longa (VLIW) 868. Em particular, a Figura 8 ilustra um exemplo de uma arquitetura de processamento gráfico. Um ou mais aspectos da arquitetura descrita em ligação com a Figura 8 podem funcionar semelhantemente aos aspectos da arquitetura Nvidia. Esta arquitetura pode prover renderização de modo imediato (EVIR) com um projeto de largura de banda elevada com o custo de energia e área de matriz. Neste projeto, o cache 801 pode ser utilizado para reduzir a largura de banda (por exemplo, para evitar amostragem repetida, leitura e/ou gravação para a memória de sistema 805). Por exemplo, o cache 801 pode ser suficientemente grande para acomodar os dados durante todo um quadro de imagem. Deve ser notado que uma ou mais dos pipelines gráficos aqui descritos, podem ter um ou mais aspectos semelhantes a pipelines DirectX e/ou OpenGL e/ou interfaces de programação de aplicações (API). Pedido de Patente US Número de Publicação 2014/0347363 descreve alguns aspectos do processamento gráfico que podem ter semelhanças com o pipeline DirectX.

[00106] O dispositivo eletrônico 866 pode incluir um circuito VLIW 868 (por exemplo, GPU) e memória de sistema

805 (por exemplo, DRAM). O circuito VLIW 868 pode incluir um pipeline gráfico 870 e memória em chip 803. O pipeline gráfico 870 pode incluir um ou mais de um front-end (FEC) 872, um coletor de vértice (VF) 874, shader de vértice (VS) 876, shader de controle de tecelagem (TCS) 878, tessellator (TESS) 880 (por exemplo, máquina de tecelagem), shader de avaliação de tecelagem (TES) 882, shader de geometria (GS) 884, rasterizador (RAST) 886, shader de fragmento (FS) 890 e unidade de saída de renderização (ROP) 894. Muitos dos rótulos de elementos foram abreviados por conveniência. A memória em chip 803 pode incluir um buffer 896, memória de composição / decomposição (C/D) 898 e cache 801. Em algumas configurações, o buffer 896 pode ser e/ou incluir um buffer de textura, buffer de quadro e/ou outro bloco de memória em chip 910. A memória de sistema 805 pode incluir dados de comando (Cmd) 807, dados de vértice 809, dados de textura e dados de visualização de acesso desordenado 811 e dados de alvos renderizados 813.

[00107] A fim de renderizar a cena, o dispositivo eletrônico 866 pode carregar dados de comando 807 e dados de vértice 809 da memória de sistema 805 para o cache 801 na memória em chip 803. Os dados de vértice 809 podem indicar um ou mais pontos (por exemplo, os vértices), as linhas e/ou polígonos. Os dados de vértice 809 podem representar uma cena e/ou um ou mais objetos em uma cena.

[00108] A FE 872 pode prover uma interface entre o cache 801 e o pipeline gráfico 870 (por exemplo, VF 874). Por exemplo, o FE 872 pode obter dados a serem desformatados pelo VF 874 e/ou pode controlar o VF 874 em relação aos dados

(por exemplo, dados de vértice) que são recuperados a partir do cache 801.

[00109] O VF 874 pode obter dados de vértice do cache 801 para o pipeline gráfico 870. O VF 874 pode acessar o buffer 896. Por exemplo, o FV pode gravar dados no buffer 896. O VF 874 pode prover os dados de vértice para o VS 876. Adicionalmente ou em alternativa, o VS 876 pode recuperar dados de vértice do buffer 896.

[00110] O VS 876 pode transformar os dados de vértice em um espaço diferente. Por exemplo, o VS 876 pode transformar os dados de vértice a partir do espaço tridimensional (3D) em um espaço bidimensional (2D). O VS 876 pode prover os dados de vértice transformados para o TCS 878 e/ou pode armazenar dados de vértice transformados no buffer 896.

[00111] O TCS 878 pode receber dados de vértice (por exemplo, dados de vértice transformados) do VS 876 e/ou do buffer 896. O TCS 878 pode transformar a base dos dados de vértice (por exemplo, os dados de vértice transformados). Por exemplo, o TCS 878 pode transformar a base dos dados de vértice em uma base de tecelagem (por exemplo, para subdividir polígonos eficientemente, tal como triângulos). O TCS 878 pode armazenar informação na memória cache 801. Por exemplo, o TCS 878 pode armazenar um indicador de base que indica a transformação de base realizada pelo TCS 878. Os dados de vértice transformados na base podem ser providos para o TESS 880.

[00112] O TESS 880 pode tecelar os dados de vértice (por exemplo, dados de vértice transformados na base). Por exemplo, o TESS 880 pode gerar mais vértices com base nos

dados de vértice (por exemplo, pode subdividir triângulos em mais triângulos). O TESS 880 pode operar com base em um parâmetro de LOD. O parâmetro de LOD pode especificar o LOD e/ou pode indicar um grau de tecelagem para o TESS 880 para executar. Os dados de vértice tecelados pode ser provida para o TES 882.

[00113] O TES 882 pode transformar a base dos dados de vértice (por exemplo, dados de vértice tecelados). Por exemplo, o TES 882 pode transformar a base de dados do vértice de volta para a base antes de a transformada de base aplicada pelo TCS 878. Em algumas configurações, o TES 882 pode ler o indicador de base a partir do cache 801, a fim de transformar (por exemplo, transformada inversa), a base dos dados de vértice. O TES 882 pode prover dados de vértice (por exemplo, os dados de vértice transformados em base inversa) para o GS 884 e/ou armazenar os dados de vértice no buffer 896.

[00114] O GS 884 pode executar uma ou mais operações sobre os dados de vértice (por exemplo, os dados de vértice transformados em base inversa). Por exemplo, o GS 884 pode gerar geometria adicional com base nos dados de vértice. O GS 884 pode prover os dados de vértice (com geometria adicional, por exemplo) para o rasterizador 886 e/ou o cache 801.

[00115] O rasterizador 886 pode receber os dados de vértice do GS 884 e/ou o cache 801. O rasterizador 886 pode rasterizar os dados de vértice. Por exemplo, o conversor pode converter os dados de vértice em dados de pixel. Os dados de pixel podem ser providos para o FS 890.

[00116] O FS 890 pode executar shader de fragmento ou pixel shader nos dados de pixel. Por exemplo, o FS 890 pode determinar um ou mais atributos (por exemplo, cor, brilho, transparência, etc.) de um ou mais pixels. Em algumas configurações, o FS 890 pode executar a remoção de superfície escondida (HSR). O FS 890 pode incluir um bloco de composição / decomposição (C/D) 892. Em alternativa, a C/D 892 pode ser separada do FS 890 e/ou acoplada ao FS 890.

[00117] Em algumas configurações, a C/D 892 pode executar operações, tais como texturização e/ou mistura. Por exemplo, a C/D 892 pode misturar várias camadas para determinar um valor final de pixel. Em algumas configurações, a C/D 892 pode combinar diferentes buffers de quadro (por exemplo, podem misturar múltiplas imagens por pixel). Deve ser notado a C/D 892 pode ser implementada no chip (no circuito VLIW 868), no pipeline gráfico 870, na memória em chip 803 e/ou separada do circuito VLIW 868. A C/D 892 não pode ser implementada na memória de sistema 805 em algumas configurações. A C/D 892 pode operar sobre dados de imagem antes dos dados da imagem serem gravados na memória de sistema 805 e/ou em uma outra a saída de fluxo (por exemplo, saída de fluxo HDMI). A FS 890 (e/ou C/D 892) pode acessar memória de C/D 898. Por exemplo, a memória de sistema 805 pode prover dados de textura e/ou dados de visualização de acesso desordenados 811 para o cache 801. O FS 890 pode realizar uma ou mais operações acessando a memória C/D e/ou cache 801. O FS 890 pode prover dados de pixel (por exemplo, dados de pixels processados) para a ROP 894.

[00118] A ROP 894 pode emitir os dados de pixels (por exemplo, dados de pixel processados). Por exemplo, a

ROP 894 pode emitir os dados de pixels através da memória C/D 898 e/ou cache 801 para memória de sistema 805 como dados de alvos renderizados 813. Os dados de alvos renderizados 813 podem ser providos (por exemplo, emitidos) para a apresentação. Por exemplo, os dados de alvos renderizados 813 podem ser enviados para um display para a apresentação.

[00119] Deve ser notado que, sem os sistemas e métodos aqui divulgados, as operações descritas em ligação com a Figura 8 podem ser realizadas em termos de conjuntos de dados (por exemplo, conjuntos de vértice, dados de pixels e/ou imagem), em que cada conjunto de dados pode ter um ou mais parâmetros estabelecidos para renderização (por exemplo, parâmetro (s) de renderizar). Por exemplo, cada conjunto de dados pode ter um LOD fixo atribuído para operações de tecelagem. Por conseguinte, LOD não pode ser alterado durante o processamento de um conjunto de dados (por exemplo, durante uma chamada de desenho). Isto pode resultar em múltiplas passagens para renderizar múltiplos LODs como descrito em ligação com as figuras 6-7, por exemplo. Adicionalmente ou em alternativa, um buffer de quadro (que pode ser implementado na memória em chip 803, buffer 896 e/ou cache 801) pode ser uniforme. Por exemplo, o buffer de quadro pode incluir dados uniformes (por exemplo, dados com o mesmo LOD, dados com o mesmo número de amostras de pixel, dados da mesma sub-rotina de shader, etc.). Por exemplo, os subconjuntos do buffer não podem ser acessados de forma independente (por exemplo, ao mesmo tempo por diferentes sub-rotinas de shader). Por exemplo, o buffer de quadro pode ser uniforme em um sentido que apenas uma gravação de dados para o buffer de quadro pode ocorrer ao

mesmo tempo. Adicionalmente ou em alternativa, o buffer de quadro pode ser uniforme no sentido de que cada gravação de dados só pode ser de uma única sub-rotina de shader de cada vez. Adicionalmente ou em alternativa, o buffer de quadro pode ser uniforme no sentido de que cada gravação de dados apenas pode incluir dados com o mesmo nível de detalhe e/ou com o mesmo número de amostras de pixel.

[00120] A figura 9 é um diagrama de blocos que ilustra um exemplo dos sistemas e processos divulgados aqui implementados na arquitetura descrita em ligação com a Figura 8. Especificamente, a Figura 9 ilustra um dispositivo eletrônico 902. O dispositivo eletrônico 902 descrito em ligação com a Figura 9 pode ser um exemplo de um ou mais dos dispositivos eletrônicos 102, 302 descritos acima em ligação com uma ou mais das Figuras 1-5. O dispositivo eletrônico 902 pode ser configurado para executar um ou mais dos métodos, funções, etapas, operações, etc., descritos em ligação com uma ou mais das Figuras 1-5.

[00121] O dispositivo eletrônico 902 pode incluir um circuito VLIW 904 (por exemplo, uma GPU) e memória de sistema 914. O circuito VLIW 904 pode ser um exemplo de um ou mais dos circuitos VLIW 104, 304 descritos acima em conexão com uma ou mais das Figuras 1-5, Adicionalmente ou em alternativa, a memória de sistema 914 pode ser um exemplo de uma ou mais das memórias de sistema 114, 314 descritas em ligação com uma ou mais das Figuras 1-5. A memória de sistema 914 pode incluir dados de comando (TMC) 907, dados de vértice 909, dados de visualização de acesso de textura e não ordenados 911 e/ou dados de destino renderizados 913. Um ou mais dos dados de comando (Cmd) 907, dados de vértice 909,

dados de visualização de acesso de textura e não ordenados 911 e/ou dados de alvos renderizados 913 podem ser semelhantes aos dados descritos em ligação com a Figura 8 correspondente.

[00122] O circuito VLIW 904 pode incluir um pipeline gráfico 906 e memória em chip 910. O pipeline gráfico 906 descrito em ligação com a Figura 9 pode ser um exemplo de um ou mais do pipelines gráficos 106, 306 descritos em ligação com uma ou mais das Figuras 1-5. Adicionalmente ou em alternativa, a memória em chip 910 descrita em ligação com a Figura 9, pode ser um exemplo de uma ou mais das memórias em chip 110, 310 descritas em ligação com uma ou mais das Figuras 1-5.

[00123] A memória em chip 910 pode incluir uma memória em buffer 996, cache 901 e/ou C/D 998. Em algumas configurações, uma ou mais da memória em chip 910, buffer 996, cache 901 e/ou memória C/D 998 podem ser semelhantes aos elementos descritos em ligação com a Figura 8. Deve ser notado que a memória em chip 910 pode incluir um ou mais buffers de quadro, como descrito em ligação com uma ou mais das Figuras 3-4. O buffer de quadro (s) pode ser incluído em uma ou mais da memória em chip 910, buffer 996, cache 901 e/ou a memória C/D 998. Adicionalmente ou em alternativa, o buffer (s) de quadro pode ser incluído na memória de sistema 914. Em algumas configurações, um ou mais buffers de quadro podem ser incluídos na memória em chip 910, memória de sistema 914, ou ambos.

[00124] O pipeline gráfico 906 pode incluir um ou mais de um FE 972, VF 974, VS 976, TCS 978, TESS 980, TES 982, GS 984, rasterizador 986, FS 990 e ROP 994. Em algumas

configurações, um ou mais de FE 972, VF 974, VS 976, TCS 978, TESS 980, TES 982, GS 984, rasterizador 986, FS 990 e ROP 994 podem executar uma ou mais das operações de um ou mais elementos correspondentes descritos em conexão com Figura 8.

[00125] Nas configurações ilustradas na Figura 9, o circuito VLIW 904 (por exemplo, pipeline gráfico 906) pode incluir um controlador de memória assíncrona 908. O controlador de memória assíncrona 908 pode ser incluído e/ou aplicado em um ou mais de TCS 978, TESS 980, TES 982, GS 984, rasterizador 986 e FS 990. Completa ou parcialmente, por exemplo, um controlador de memória assíncrona 908A pode ser implementado no TCS 978, no TESS 980, no TES 982, no GS 984, no rasterizador 986 e/ou no FS 990. O rótulo genérico "908" pode indicar que o controlador de memória assíncrona 908 pode ser implementado em qualquer um ou todos os elementos 908a-f ilustrados na Figura 9. Em algumas configurações, o controlador de memória assíncrona 908 pode ser implementado apenas em um ou mais de TCS 978, TESS 980, TES 982 e GS 984. Por exemplo, o controlador de memória assíncrona 908 só pode ler assincronamente um ou mais parâmetros no TCS 978, TESS 980, TES 982 e/ou GS 984, uma vez que os dados podem ser transmitidos para as fases posteriores.

[00126] O controlador de memória assíncrona 908 descrito em ligação com a Figura 9, pode ser um exemplo de um ou mais dos controladores de memória assíncrona 108, 308 descritos em ligação com uma ou mais das Figuras 1-5. Por exemplo, o controlador de memória assíncrona 908 podem pode de forma assíncrona ler um ou mais parâmetros (por exemplo,

parâmetros de LOD) a partir da memória em chip 910 e/ou da memória de sistema 914 (durante o processamento de um conjunto de dados, por exemplo). Em algumas configurações, o parâmetro (s) atualizado pode ser provido a um ou mais do TCS 978, do TESS 980, do TES 982, do GS 984, do rasterizador 986 e do FS 990. Deste modo, um ou mais dos TCS 978, do TESS 980, do TES 982, do GS 984, do rasterizador 986 e do FS 990 podem alterar o processamento do conjunto de dados para ser diferente do processamento inicialmente estabelecido para um conjunto de dados, enquanto o conjunto de dados está sendo processado.

[00127] Em algumas configurações, isto pode permitir que o dispositivo eletrônico 902 execute renderização de única passagem multi-LOD. Por exemplo, o controlador de memória assíncrona 908 pode ler um parâmetro de LOD que é diferente de um outro parâmetro de LOD inicialmente estabelecido para o conjunto de dados que está atualmente sendo processado e/ou renderizado (por exemplo, em tecelado). O parâmetro de LOD atualizado pode ser provido a um ou mais do TCS 978, do TESS 980, do TES 982, do GS 984, do rasterizador 986 e do FS 990. Deste modo, um ou mais do TCS 978, do TESS 980, do TES 982, do GS 984, do rasterizador 986 e do FS 990 podem mudar processamento (por exemplo, mudar nível de tecelagem) de um conjunto de dados de acordo com o parâmetro de LOD atualizado durante o processamento do conjunto de dados que foi iniciado com um ou mais diferentes parâmetros.

[00128] O circuito VLIW 904 (por exemplo, pipeline gráfico 906) pode incluir um controlador de buffer de quadro não uniforme 912. Tal como ilustrado na Figura 9, o

controlador de buffer de quadro não uniforme 912 pode ser incluído e/ou implementado no FS 990 (por exemplo, a C/D 992). Deve ser notado que o controlador de buffer de quadro não uniforme 912 pode ser adicionalmente ou em alternativa incluído e/ou implementado em um ou mais do FS 990, da C/D 992, pipeline gráfico 906 e memória em chip 910. Completa ou parcialmente, por exemplo, um controlador de buffer de quadro não uniforme 912 pode ser implementado em um ou mais do FS 990, da C/D 992, do pipeline gráfico 906, da memória em chip 910 e do dispositivo eletrônico 902. O controlador de buffer de quadro não uniforme 912 pode ser implementado na C/D 992. Deve ser notado que a latência adicional pode ser efetuada de outra maneira.

[00129] O controlador de buffer de quadro não uniforme 912 descrito em ligação com a Figura 9, pode ser um exemplo de um ou mais dos controladores de buffer de quadro não uniformes 112, 312 descritos em ligação com uma ou mais das Figuras 1-5. Por exemplo, o controlador de buffer de quadro não uniforme 912 pode acessar dinamicamente diferentes subconjuntos de um ou mais buffer (s) de quadro. Em algumas configurações, o acesso dinamicamente diferente de subconjuntos do buffer de quadro pode incluir o acesso a pelo menos um pixel de uma primeira zona a uma taxa diferente de pelo menos um pixel de uma segunda zona. Adicionalmente ou em alternativa, o acesso dinamicamente diferente de subconjuntos do buffer de quadro pode incluir permitir a troca de sub-rotina de shader dinâmica (quando o processamento de um primeiro pixel está em andamento e o processo de um segundo pixel é finalizado, por exemplo). Adicionalmente ou em alternativa, o controlador de buffer de

quadro não uniforme 912 pode ser configurado para substituir uma primeira amostra processada de um primeiro pixel por uma segunda amostra de um segundo pixel para evitar o processamento da segunda amostra do segundo pixel.

[00130] Em algumas configurações, um cache dedicado pode ser implementado para lidar com texturas específicas e renderizar alvos. O cache pode ser tão grande como o conjunto das próprias texturas.

[00131] A figura 10 é um diagrama de blocos que ilustra um outro exemplo de um dispositivo eletrônico 1066 que inclui uma circuito de palavra de instrução muito longa (VLIW) 1068. Em particular, a Figura 10 ilustra um outro exemplo de uma arquitetura de processamento gráfico. Um ou mais aspectos da arquitetura descrita em ligação com a Figura 10 pode funcionar semelhantemente aos aspectos da arquitetura Intel. Esta arquitetura pode prover renderização de modo imediata (EVIR), bem como uma arquitetura de memória unificada (UMA) com memória física e virtual compartilhada. Esta arquitetura pode permitir o compartilhamento de cópia zero, bem como o compartilhamento de ponteiro.

[00132] O dispositivo eletrônico 1066 pode incluir um processador 1015 (por exemplo, uma CPU). Neste exemplo, o processador 1015 pode incluir um circuito VLIW 1068. Por exemplo, o circuito VLIW 1068 pode ser integrado no processador 1015.

[00133] O dispositivo eletrônico 1066 pode incluir um circuito VLIW 1068 (por exemplo, GPU) e memória de sistema 1005. O circuito VLTW 1068 pode incluir um pipeline gráfico 1070 e memória em chip 1003. O pipeline gráfico 1070 pode incluir um ou mais de uma streamer de comando (CS) 1017, um

coletor de vértice (VF) 1074, shader de vértice (VS) 1076, hull shader (HS) 1078, tessellator (TESS) 1080 (por exemplo, máquina de tecelagem), shader de domínio (DS) 1082, shader de geometria (GS) 1084, saída de fluxo (SOL) 1019, clipper (Clip) 1021, rasterizador (Rast) 1086 e shader de fragmento (FS) 1090. Muitos dos rótulos de elementos foram abreviados por conveniência. A memória em chip 1003 pode incluir um ou mais buffers, um ou mais caches, etc. Em algumas configurações, a memória em chip 1003 pode incluir um buffer de textura, buffer de quadro e/ou um outro bloco de memória em chip 1110. A memória de sistema 1005 pode incluir dados de comando, dados de vértice (por exemplo, dados de geometria) e/ou dados de textura.

[00134] A fim de renderizar a cena, o dispositivo eletrônico 1066 pode obter dados de comando e dados de vértice a partir da memória de sistema. Por exemplo, o CS 1017 pode receber e/ou executar comando (s) de renderização a partir da memória de sistema 1005. Os dados de vértice podem indicar um ou mais pontos (por exemplo, os vértices), as linhas e/ou polígonos. Os dados de vértice 1009 pode representar uma cena e/ou um ou mais objetos em uma cena.

[00135] O VF 1074 pode buscar dados de vértice da memória de sistema 1005 para o circuito VLIW 1068 (por exemplo, pipeline gráfico 1070). O VF 1074 pode prover os dados de vértice para o VS 1076 e/ou pode armazenar os dados de vértice no cache 1001.

[00136] O VS 1076 pode transformar os dados de vértice em um espaço diferente. Por exemplo, o VS 1076 pode transformar os dados de vértice do espaço tridimensional (3D) em um espaço bidimensional (2D). O VS 1076 pode prover

os dados de vértice transformados para o HS 1078 e/ou pode armazenar dados de vértice transformados no cache 1001.

[00137] O HS 1078 pode receber dados de vértice (por exemplo, dados de vértice transformados) do VS 1076 e/ou a partir do cache 1001. O HS 1078 pode transformar a base dos dados de vértice (por exemplo, os dados de vértice transformados). Por exemplo, o HS 1078 pode transformar a base dos dados de vértice em uma base para tecelagem (por exemplo, para eficientemente subdividir polígonos, tal como triângulos). O HS 1078 pode armazenar informação no cache 1001. Por exemplo, o HS 1078 pode armazenar um indicador de base que indica a transformação de base realizada pelo HS 1078. Os dados de vértice transformados na base pode ser providos para o TESS 1080.

[00138] O TESS 1080 pode tecelar os dados de vértice (por exemplo, dados de vértice transformados na base). Por exemplo, o TESS 1080 pode gerar mais vértices com base nos dados de vértice (por exemplo, pode subdividir triângulos em mais triângulos). O TESS 1080 pode operar com base em um parâmetro de LOD. O parâmetro de LOD pode especificar o LOD e/ou pode indicar um grau de tecelagem para o TESS 1080 a executar. Os dados de vértice tecelados podem ser providos para o DS 1082.

[00139] O DS 1082 pode transformar a base dos dados de vértice (por exemplo, dados de vértice tecelados). Por exemplo, o DS 1082 pode transformar a base de dados do vértice de volta para a base antes da transformada de base aplicada pelo SH 1078. Em algumas configurações, o DS 1082 pode ler o indicador de base a partir do cache 1001, a fim de transformar (por exemplo, transformada inversa) a base

dos dados de vértice. Em outras configurações, o DS 1082 pode ser acoplado diretamente ao HS 1078 e pode receber o indicador de base a partir do HS 1078, a fim de transformar (por exemplo, transformada inversa) a base dos dados de vértice. O DS 1082 pode prover dados de vértice (por exemplo, os dados de vértice transformados na base inversa) para o GS 1084 e/ou armazenar os dados de vértice no cache 1001.

[00140] O GS 1084 pode executar uma ou mais operações sobre os dados de vértice (por exemplo, os dados de vértice transformados na base inversa). Por exemplo, o GS 1084 pode gerar geometria adicional com base nos dados de vértice. O GS 1084 pode prover os dados de vértice (com geometria adicional, por exemplo) para o SOL 1019.

[00141] O SOL 1019 pode transmitir os dados de vértice (com geometria adicional, por exemplo) e/ou armazenar os dados de vértice na memória de sistema 1005. O SOL 1019 pode prover os dados de vértice (com geometria adicional, por exemplo) para o clipper 1021.

[00142] O clipper 1021 pode descartar alguns dos dados de vértice (com geometria adicional, por exemplo). Por exemplo, o clipper 1021 pode descartar dados de vértice que se encontram do lado de fora de um ponto de visualização (por exemplo, frustum de visualização). O clipper 1021 pode prover os dados de vértice cortados para o rasterizador 1086 e/ou o cache 1001.

[00143] O rasterizador 1086 pode receber os dados de vértice cortados a partir do clipper 1021 e/ou do cache 1001. O rasterizador 1086 pode rasterizar os dados de vértice. Por exemplo, o rasterizador pode converter os dados

de vértice em dados de pixel. Os dados de pixel podem ser providos para o FS 1090 e/ou para o cache 1001.

[00144] O FS 1090 pode executar shader de fragmento ou pixel shader nos dados de pixel. Por exemplo, o FS 1090 pode determinar um ou mais atributos (por exemplo, cor, brilho, transparência, etc.) de um ou mais pixels. Em algumas configurações, o FS 1090 pode executar a remoção de superfície escondida (HSR). Em algumas configurações, o FS 1090 pode incluir um bloco de composição / decomposição (C/D) 1092. Em alternativa, a C/D 1092 pode ser separada do FS 1090 e/ou acoplada ao FS 1090 tal como ilustrado na Figura 10.

[00145] Em algumas configurações, a C/D 1092 pode executar operações tais como texturização e/ou mistura. Por exemplo, a C/D 1092 pode misturar várias camadas para determinar um valor final de pixel. Em algumas configurações, a C/D 1092 pode combinar diferentes buffers de quadro (por exemplo, pode misturar várias imagens por pixel). Deve ser notado que a C/D 1092 pode ser implementada no chip (no circuito VLIW 1068), na pipeline gráfico 1070, na memória em chip 1003 e/ou separada do circuito VLIW 1068. A C/D 1092 pode operar sobre dados de imagem antes dos dados de imagem serem gravados na memória de sistema 1005 e/ou em uma outra saída de fluxo (por exemplo, saída de fluxo HDMI). Em algumas configurações, a memória de sistema 1005 pode prover dados de textura para o cache 1001, que pode ser acessado pelo FS 1090 e/ou pela C/D 1092. O FS 1090 pode realizar uma ou mais operações acessando o cache 1001. O FS 1090 pode prover os dados de pixel (por exemplo, dados de pixel processados)

para a C/D 1092, que pode realizar composição / decomposição e prover os dados de pixel para o cache 1001.

[00146] Os dados de pixel podem ser emitidos através do cache 1001 para a memória de sistema 1005. Os dados de pixel podem ser provados (por exemplo, emitidos) para apresentação. Por exemplo, os dados de pixel podem ser enviados para uma exibição para a apresentação.

[00147] Deve ser notado que, sem os sistemas e métodos aqui divulgados, as operações descritas em ligação com a Figura 10 podem ser realizadas em termos de conjuntos de dados (por exemplo, conjuntos de vértice, dados de pixels e/ou de imagem), como descrito em conexão com a Figura 8.

[00148] A figura 11 é um diagrama de blocos que ilustra um outro exemplo dos sistemas e processos divulgados aqui implementados na arquitetura descrita em ligação com a Figura 10. Especificamente, a Figura 11 ilustra um dispositivo eletrônico 1102. O dispositivo eletrônico 1102 descrito em conexão com a Figura 11 pode ser um exemplo de um ou mais dos dispositivos eletrônicos 102, 302 descritos acima em ligação com uma ou mais das Figuras 1-5. O dispositivo eletrônico 1102 pode ser configurado para executar um ou mais dos métodos, funções, etapas, operações, etc., descritos em ligação com uma ou mais das Figuras 1-5.

[00149] O dispositivo eletrônico 1102 pode incluir um processador 1124 (por exemplo, uma CPU). O processador 1124 pode ser um exemplo do processador 324 descrito em ligação com a Figura 3. Neste exemplo, o processador 1015 pode incluir um circuito VLIW 1104. Por exemplo, o circuito VLIW 1104 pode ser integrado no processador 1124.

[00150] O dispositivo eletrônico 1102 pode incluir um circuito VLIW 1104 (por exemplo, uma GPU) e memória de sistema 1114. O circuito VLIW 1104 pode ser um exemplo de um ou mais dos circuitos VLIW 104, 304 descritos acima em conexão com uma ou mais das Figuras 1-5, Adicionalmente ou em alternativa, a memória de sistema 1114 pode ser um exemplo de uma ou mais das memórias de sistema 114, 314 descritas em ligação com uma ou mais das Figuras 1-5. A memória de sistema 1114 pode incluir dados de comando, dados de vértice (por exemplo, dados de geometria), dados de textura e/ou dados de alvos renderizados (por exemplo, dados de pixel).

[00151] O circuito VLIW 1104 pode incluir um pipeline gráfico 1106 e memória em chip 1110. O pipeline gráfico 1106 descrito em ligação com a Figura 11 pode ser um exemplo de um ou mais dos pipelines gráficos 106, 306 descritos em ligação com uma ou mais das Figuras 1-5. Adicionalmente ou em alternativa, a memória em chip 1110 descrita em conexão com a Figura 11 pode ser um exemplo de uma ou mais das memórias em chip 110, 310 descritas em ligação com uma ou mais das Figuras 1-5.

[00152] A memória em chip 1110 pode incluir um ou mais buffers de quadro, um ou mais cache 1101, etc. Por exemplo, caches de nível 1 (L1) e/ou nível 2 (L2) podem ser utilizados como armazenamento em chip. Em algumas configurações, hardware predição de pipeline pode ser desativado. Deve ser notado que a memória em chip 1110 pode incluir um ou mais buffers de quadro, como descrito em ligação com uma ou mais das Figuras 3-4. O buffer (s) de quadro pode ser incluído em uma ou mais da memória em chip 1110, no cache 1101, etc. Adicionalmente ou em alternativa,

- buffer (s) de quadro pode ser incluído na memória de sistema 1114. Em algumas configurações, um ou mais buffers de quadro podem ser incluídos na memória em chip 1110, na memória de sistema 1114 ou ambas.

[00153] O pipeline gráfico 1106 pode incluir um ou mais de um CS 1117, VF 1174, VS 1176, HS 1178, TESS 1180, DS 1182, GS 1184, SOL 1119, clipper 1121, rasterizador 1186, SF 1190 e C/D 1192. Em algumas configurações, um ou mais de CS 1117, VF 1174, VS 1176, HS 1178, TESS 1180, DS 1182, GS 1184, SOL 1119, clipper 1121, rasterizador 1186, SF 1190 e C/D 1192 podem executar uma ou mais das operações de um ou mais elementos correspondentes descritos em ligação com a Figura 10.

[00154] Nas configurações ilustradas na Figura 11, o circuito VLIW 1104 (por exemplo, pipeline gráfico 1106) pode incluir um controlador de memória assíncrona 1108. O controlador de memória assíncrona 1108 pode ser incluído e/ou aplicado em um ou mais do HS 1178, TESS 1180, DS 1182, GS 1184, rasterizador e 1186 e FS 1190. Completa ou parcialmente, por exemplo, um controlador de memória assíncrona 1108a pode ser implementado no SH 1178, TESS 1180, o DS 1182, o GS 1184, o rasterizador 1186 e/ou o FS 1190. O rótulo genérico "1108" pode indicar que o controlador de memória assíncrona 1108 pode ser implementado em qualquer um ou todos os elementos 1108a-f ilustrados na Figura 11. Em algumas configurações, o controlador de memória assíncrona 1108 pode ser implementado apenas em um ou mais do HS 1178, TESS 1180, DS 1182 e GS 1184. Por exemplo, o controlador de memória assíncrona 1108 só pode ler de forma assíncrona um ou mais parâmetros do HS 1178, TESS 1180, DS 1182 e/ou GS

1184, desde que os dados possam ser encaminhados para estágios posteriores.

[00155] O controlador de memória assíncrona 1108 descrito em conexão com a Figura 11, pode ser um exemplo de um ou mais dos controladores de memória assíncrona 108, 308 descritos em ligação com uma ou mais das Figuras 1-5. Por exemplo, o controlador de memória assíncrona 1108 pode ler de forma assíncrona um ou mais parâmetros (por exemplo, parâmetros de LOD) a partir da memória em chip 1110 e/ou da memória de sistema 1114 (durante o processamento de um conjunto de dados, por exemplo). Em algumas configurações, o parâmetro (s) atualizado pode ser provido a um ou mais do HS 1178, TESS 1180, o DS 1182, o GS 1184, o rasterizador 1186 e o SF 1190. Por conseguinte, um ou mais do HS 1178, o TESS 1180, o DS 1182, o GS 1184, o rasterizador, e o FS. 1190. Assim, um ou mais do HS 1178, o TESS 1180, o DS 1182, o GS 1184, o rasterizador 1186 e o FS 1190 podem alterar o processamento do conjunto de dados para ser diferente do processamento inicialmente estabelecido para um conjunto de dados, enquanto o conjunto de dados está sendo processado.

[00156] Em algumas configurações, isto pode permitir que o dispositivo eletrônico 1102 execute uma renderização de única passagem multi-LOD. Por exemplo, o controlador de memória assíncrona 1108 pode ler um parâmetro de LOD que é diferente de um outro parâmetro de LOD inicialmente estabelecido para o conjunto de dados que está atualmente sendo processado e/ou fundido (por exemplo, em xadrez). O parâmetro de LOD atualizado pode ser provido a um ou mais do HS 1178, TESS 1180, o DS 1182, o GS 1184, o rasterizador 1186 e o SF 1190. Por conseguinte, um ou mais

do HS 1178, TESS 1180, o DS 1182, o GS 1184, o rasterizador 1186 e o SF 1190 pode mudar de processamento (por exemplo, mudar nível de tecelagem) de um conjunto de dados de acordo com o parâmetro de LOD atualizado durante o processamento do conjunto de dados, que foi iniciado com um ou mais diferentes parâmetros.

[00157] O circuito VLIW 1104 (por exemplo, pipeline gráfico 1106) pode incluir um controlador de buffer de quadro não uniforme 1112. Como ilustrado na Figura 11, o controlador de buffer de quadro não uniforme 1112 pode ser incluído e/ou implementado na C/D 1192. Deve ser notado que o controlador de buffer de quadro não uniforme 1112 pode ser adicionalmente ou em alternativa incluído e/ou implementado em um ou mais do FS 1190, C/D 1192, pipeline gráfico 1106 e memória em chip 1110. Completa ou parcialmente, por exemplo, um controlador de buffer de quadro não uniforme 1112 pode ser implementado em um ou mais do FS 1190, C/D 1192, pipeline gráfico 1106, memória em chip 1110 e o dispositivo eletrônico 1102.

[00158] O controlador de buffer de quadro não uniforme 1112 descrito em conexão com a Figura 11, pode ser um exemplo de um ou mais dos controladores buffer de quadro não uniformes 112, 312 descritos em ligação com uma ou mais das Figuras 1-5. Por exemplo, o controlador de buffer de quadro não uniforme 1112 pode acessar dinamicamente diferentes subconjuntos de um ou mais buffer (s) de quadro. Em algumas configurações, o acesso dinamicamente diferente de subconjuntos do buffer de quadro pode incluir o acesso a pelo menos um pixel de uma primeira zona a uma taxa diferente de pelo menos um pixel de uma segunda zona. Adicionalmente

ou em alternativa, o acesso dinamicamente diferente de subconjuntos do buffer de quadro pode incluir permitir troca de sub-rotina de shader dinâmico (quando o processamento de um primeiro pixel está em andamento e processo de um segundo pixel é finalizado, por exemplo). Adicionalmente ou em alternativa, o controlador de buffer de quadro não uniforme 1112 pode ser configurado para substituir uma primeira amostra processada de um primeiro pixel para uma segunda amostra de um segundo pixel para evitar o processamento da segunda amostra do segundo pixel. Deve ser notado que, em algumas configurações, o dispositivo eletrônico 1102 descrito em conexão com a Figura 11 pode ser implementado para realizar antialiasing adaptativo assíncrono usando memória compartilhada e/ou shader multifreqüência adaptativo.

[00159] A figura 12 é um diagrama de blocos que ilustra um outro exemplo de um dispositivo eletrônico 1266 que inclui uma circuito de palavra de instrução muito longa (VLIW) 1268. Em particular, a Figura 12 ilustra um exemplo de uma arquitetura de processamento gráfico. Um ou mais aspectos da arquitetura descrita em ligação com a Figura 12 pode funcionar semelhante aos aspectos da arquitetura Imagine Technologies (por exemplo, gráficos VRPower). Esta arquitetura é uma arquitetura de mosaicos que pode prover Renderização Diferida Baseada em Mosaico (TBDR). Geometria opaca pode ser classificada em mosaicos (em vez de cache, por exemplo).

[00160] O dispositivo eletrônico 1266 pode incluir um circuito VLIW 1268 (por exemplo, GPU) e a memória de sistema 1205 (por exemplo, DRAM). O circuito VLIW 1268 pode

incluir um pipeline gráfico 1270 e memória em chip 1203. O pipeline gráfico 1270 pode incluir um ou mais de um processador de vértice (VP) 1274, shader de vértice (VS) 1276, shader de controle de tecelagem (TCS) 1278, tessellator (TESS) 1280 (por exemplo, máquina de tecelagem), shader de avaliação de tecelagem (TES) 1282, shader de geometria (GS) 1284, mosaico 1223, rasterizador (RAST) 1286 e shader de fragmento (FS) 1290. Muitos dos rótulos de elemento têm sido abreviados por conveniência. A memória em chip 1203 pode incluir um buffer de profundidade 1237 e um buffer de cor 1239. A memória de sistema 1205 pode incluir dados de geometria 1209, um buffer de parâmetro 1235, dados de textura 1211 e um buffer de quadro 1241. Os dados de geometria 1209 podem indicar um ou mais pontos (por exemplo, vértices), linhas e/ou polígonos. Os dados de geometria 1209 podem representar uma cena e/ou um ou mais objetos em uma cena.

[00161] O VP 1274 pode ir buscar dados de geometria da memória de sistema 1205 para o circuito VLIW 1268 (por exemplo, pipeline gráfico 1270). O VP 1274 pode prover os dados de geometria para o VS 1276.

[00162] O VS 1276 pode transformar (por exemplo, os projetar) dados de geometria em um espaço diferente. Por exemplo, o VS 1276 pode transformar os dados de geometria do espaço tridimensional (3D) em um espaço bidimensional (2D). Em algumas configurações, o VS 1276 pode executar recorte, projeção e/ou descarte. Por exemplo, o VS 1276 pode recortar dados de geometria que estão fora do volume de visualização. O VS 1276 pode projetar os dados de geometria do espaço 3D no espaço 2D. O VS 1276 também pode abater alguns dos dados de geometria. Por exemplo, os dados de geometria

correspondentes à geometria ocluída (por exemplo, a geometria que não é visível porque está bloqueada por outra geometria) podem ser removidos. O VS 1276 pode prover os dados de geometria transformados para o TCS 1278.

[00163] O TCS 1278 pode receber dados de geometria (por exemplo, dados de geometria transformados) do VS 1276 e/ou a partir do buffer 1296. O TCS 1278 pode transformar a base dos dados de geometria (por exemplo, os dados de geometria transformados). Por exemplo, o TCS 1278 pode transformar a base dos dados de geometria para uma base de tecelagem (por exemplo, para subdividir eficientemente polígonos, tal como triângulos). Por exemplo, o TCS 1278 pode produzir um indicador de base que indica a transformação de base realizada pelo TCS 1278. Os dados de geometria transformados na base podem ser provados para o TESS 1280.

[00164] O TESS 1280 pode tecelar os dados geométricos (por exemplo, dados de geometria transformados na base). Por exemplo, o TESS 1280 pode gerar mais dados de geometria (por exemplo, os vértices) com base nos dados de geometria (por exemplo, pode subdividir triângulos em mais triângulos). O TESS 1280 pode operar com base em um parâmetro de LOD. O parâmetro de LOD pode especificar o LOD e/ou pode indicar um grau de tecelagem para o TESS 1280 para executar. Os dados de geometria tecelados podem ser provados para o TES 1282.

[00165] O TES 1282 pode transformar a base dos dados de geometria (por exemplo, dados de geometria tecelados). Por exemplo, o TES 1282 pode transformar a base dos dados de geometria de volta para a base antes da transformada de base aplicada pelo TCS 1278. Em algumas

configurações, o TES 1282 pode obter o indicador de base a partir do SCT 1278 e/ou da memória (por exemplo, memória em chip 1203 e/ou memória de sistema 1205), a fim de transformar (por exemplo, transformada inversa), a base dos dados de geometria. O TES 1282 pode prover dados de geometria (por exemplo, os dados de geometria transformados com base inversa) para o GS 1284.

[00166] O GS 1284 pode executar uma ou mais operações sobre os dados de geometria (por exemplo, os dados de geometria transformados na base inversa). Por exemplo, o GS 1284 pode gerar geometria adicional com base nos dados de geometria. O GS 1284 pode prover os dados de geometria (com geometria adicional, por exemplo) para o ladrilhador 1223.

[00167] O ladrilhador 1223 pode dividir os dados de geometria (com geometria adicional, por exemplo) em mosaicos. Por exemplo, o ladrilhador 1223 pode dividir os dados de geometria em grupos menores de dados de geometria. Os mosaicos podem incluir dados de geometria contíguos. O ladrilhador 1223 pode prover os mosaicos para o buffer de parâmetro 1235.

[00168] O buffer de parâmetro 1235 pode armazenar os mosaicos. Por exemplo, o buffer de parâmetro 1235 pode armazenar uma lista de primitivas (por exemplo, dados de geometria, vértices, linhas, polígonos, etc.). Na arquitetura de mosaicos descrita na Figura 12, o processamento de mosaico pode alavancar localidade espacial de acessos de memória de sistema. Por exemplo, operações que são locais para um mosaico podem ser relativamente "baratas" (por exemplo, pode utilizar muito pouco processamento, energia, memória e/ou recursos de tempo). Em algumas

configurações, o circuito VLIW 1268 pode tecelar os dados de geometria e enviar os dados de geometria para fora para mosaicos para o buffer de parâmetro 1235. Os dados de geometria (por exemplo, dados de geometria em mosaico, primitivos, etc.) podem ser providos para o rasterizador 1286. Deve ser notado que os parâmetros armazenados no buffer de parâmetro 1235 pode não incluir parâmetro (s) de renderização (por exemplo, parâmetro de LOD (s)).

[00169] O rasterizador 1286 pode receber os dados de geometria (por exemplo, dados de geometria em mosaico, primitivos, etc.) a partir do buffer de parâmetro 1235. O rasterizador 1286 pode rasterizar os dados de geometria. Por exemplo, o rasterizador pode converter os dados de geometria em dados de pixel. Os dados de pixel podem ser providos ao FS 1290.

[00170] O FS 1290 pode executar shader de fragmento ou pixel shader nos dados de pixel. Por exemplo, o FS 1290 pode determinar um ou mais atributos (por exemplo, cor, brilho, transparência, etc.) de um ou mais pixels. Em algumas configurações, o FS 1290 pode incluir um ou mais de uma remoção de bloco de superfície escondida (HSR) e teste de profundidade (DT) 1225, um buffer de marcação (TB) 1227, uma bloco de textura e shade (T&S) 1229, um bloco de teste alfa (AT) 1231 e um bloco de mistura alfa (AB) 1233. O HSR & DT 1225 pode executar testes de profundidade e remoção de superfície escondida. Por exemplo, o teste de profundidade pode incluir a determinação de uma profundidade correspondente a pixels. O HSR & DT pode determinar a profundidade de cada pixel e armazenar indicadores de profundidade no buffer de profundidade 1237. Realizar

remoção de superfície escondida pode incluir descartar os pixels que estão a uma profundidade maior do que os outros pixels (por exemplo, que são obstruídos por outros pixels). Em algumas abordagens, o buffer de marcação (TB) 1227 pode opcionalmente reagrupar pixels para uma execução mais rápida. A T & S 1229 pode obter dados de textura 1211 da memória de sistema. O AT 1231 pode realizar um teste "stencil" personalizado (por exemplo, um outro teste de profundidade). Em algumas configurações, o AT 1231 pode atualizar o HSR & DT 1225. O AB 1233 pode misturar pixels (por exemplo, os pixels de uma ou mais camadas e/ou buffers de quadro). Por exemplo, o AB 1233 pode realizar uma operação de composição e/ou de decomposição, em que duas imagens podem ser misturadas usando um valor alfa.

[00171] O AB 1233 pode prover dados de pixel para o buffer de cores 1239. Como ilustrado na Figura 12, os dados de buffer de cor e de profundidade podem ser armazenados no chip. O buffer de cor 1239 pode prover os dados de pixel para o buffer de quadro 1241. O buffer de quadro 1241 pode emitir os dados de pixel. Os dados de pixel podem ser provados (por exemplo, emitidos) para a apresentação. Por exemplo, os dados de pixel podem ser enviados para uma exibição para a apresentação.

[00172] Deve ser notado que, sem os sistemas e métodos aqui divulgados, as operações descritas em ligação com a Figura 12 podem ser realizadas em termos de conjuntos de dados (por exemplo, conjuntos de vértice, dados de pixels e/ou de imagem), como descrito em conexão com a Figura 8.

[00173] A figura 13 é um diagrama de blocos que ilustra um exemplo dos sistemas e métodos aqui descritos

implementados na arquitetura descrita em ligação com a Figura 12. Esta arquitetura pode prover TBDR. Especificamente, a Figura 13 ilustra um dispositivo eletrônico 1302. O dispositivo eletrônico 1302 descrito em conexão com a Figura 13 pode ser um exemplo de um ou mais dos dispositivos eletrônicos 102, 302 descritos acima em ligação com uma ou mais das Figuras 1-5. O dispositivo eletrônico 1302 pode ser configurado para executar um ou mais dos métodos, funções, etapas, operações, etc., descritos em ligação com uma ou mais das Figuras 1-5.

[00174] O dispositivo eletrônico 1302 pode incluir um circuito VLIW 1304 (por exemplo, uma GPU) e memória de sistema 1314. O circuito VLIW 1304 pode ser um exemplo de um ou mais dos circuitos VLIW 104, 304 descritos acima em conexão com uma ou mais das Figuras 1-5, Adicionalmente ou em alternativa, a memória de sistema 1314 pode ser um exemplo de uma ou mais das memórias de sistema 114, 314 descritas em ligação com uma ou mais das Figuras 1-5. A memória de sistema 1314 pode incluir dados de geometria 1309, um buffer de parâmetro 1335, dados de textura 1311 e/ou um buffer de quadro 1341. Um ou mais dos dados de geometria 1309, um buffer de parâmetro 1335, dados de textura 1311 e/ou um buffer de quadro 1341 podem ser semelhantes aos elementos descritos em ligação com a Figura 12 correspondente. Deve ser notado que a memória de sistema 1314 pode incluir um ou mais buffers de quadro, como descrito em ligação com uma ou mais das Figuras 3-4. Por exemplo, o buffer de quadro 1341 pode ser um exemplo de um ou mais dos buffers de quadro descritos em ligação com uma ou mais das Figuras 3-4. Em algumas configurações, um ou mais buffers de quadro podem

ser incluídos na memória em chip 1310, a memória de sistema 1314 ou ambas.

[00175] O circuito VLIW 1304 pode incluir um pipeline gráfico 1306 e memória em chip 1310. O pipeline gráfico 1306 descrito em ligação com a Figura 13 pode ser um exemplo de um ou mais dos pipelines gráficos 106, 306 descritos em ligação com uma ou mais das Figuras 1-5. Adicionalmente ou em alternativa, a memória em chip 1310 descrita em conexão com a Figura 13, pode ser um exemplo de uma ou mais das memórias em chip 110, 310 descritas em ligação com uma ou mais das Figuras 1-5.

[00176] A memória em chip 1310 pode incluir um cache 1301 e/ou um buffer de profundidade e de cor 1343. Em algumas configurações, um ou mais da memória em chip 1310, do cache 1301 e/ou do buffer de profundidade e de cor 1343 podem ser semelhantes aos elementos descritos em ligação com a Figura 12 correspondente.

[00177] O pipeline gráfico 1306 pode incluir um ou mais de um VP 1374, VS 1376, TCS 1378, TESS 1380, TES 1382, GS 1384, ladrilhador 1323, 1386 rasterizador e FS 1390. Em algumas configurações, um ou mais do VP 1374, VS 1376, TCS 1378, TESS 1380, TES 1382, GS 1384, ladrilhador 1323, rasterizador 1386 e FS 1390 pode executar uma ou mais das operações de um ou mais elementos correspondentes descritos em ligação com a Figura 12. Deve ser notado que o FS 1390 pode incluir um ou mais de um HSR & DT, TB, T & S, nA e AB como descrito em conexão com a Figura 12, em algumas configurações.

[00178] Nas configurações ilustradas na Figura 13, o circuito VLIW 1304 (por exemplo, pipeline gráfico 1306)

pode incluir um controlador de memória assíncrona 1308. O controlador de memória assíncrona 1308 pode ser incluído e/ou implementado em um ou mais do TCS 1378, o TESS 1380, o TES 1382, o GS 1384, o rasterizador 1386 e o SF 1390. Completa ou parcialmente, por exemplo, um controlador de memória assíncrona 1308 pode ser implementado no TCS 1378, TESS 1380, o TES 1382, o GS 1384, o rasterizador 1386 e/ou o FS 1390. O rótulo genérico "1308" pode indicar que o controlador de memória assíncrona 1308 pode ser implementado em qualquer um ou todos os elementos 1308a-f ilustrados na Figura 13. Em algumas configurações, o controlador de memória assíncrona 1308 pode ser implementado apenas em um ou mais do TCS 1378, TESS 1380, o TES 1382 e o GS 1384. Por exemplo, o controlador de memória assíncrona 1308 pode apenas ler de forma assíncrona um ou mais parâmetros no TCS 1378, TESS 1380, TES 1382 e/ou GS 1384, uma vez que os dados podem ser encaminhados para as fases posteriores.

[00179] O controlador de memória assíncrona 1308 descrito em conexão com a Figura 13, pode ser um exemplo de um ou mais dos controladores de memória assíncrona 108, 308 descritos em ligação com uma ou mais das Figuras 1-5. Por exemplo, o controlador de memória assíncrona 1308 pode de forma assíncrona ler um ou mais parâmetros (por exemplo, parâmetros de LOD) a partir da memória de sistema 1314 (durante o processamento de um conjunto de dados, por exemplo). Em algumas configurações, o parâmetro (s) atualizado pode ser provido para um ou mais do TCS 1378, o TESS 1380, o TES 1382, o GS 1384, o rasterizador 1386 e o SF 1390. Deste modo, um ou mais do TCS 1378, TESS 1380, o TES 1382, o GS 1384, o rasterizador 1386 e o SF 1390 pode alterar

o processamento do conjunto de dados para ser diferente do processamento inicialmente estabelecido para um conjunto de dados, enquanto o conjunto de dados está sendo processado.

[00180] Em algumas configurações, isto pode permitir que o dispositivo eletrônico 1302 execute renderização de única passagem multi-LOD. Por exemplo, o controlador de memória assíncrona 1308 pode ler um parâmetro de LOD que é diferente de um outro parâmetro de LOD inicialmente estabelecido para o conjunto de dados que está atualmente sendo processado e/ou renderizado (por exemplo, tecelado). O parâmetro de LOD atualizado pode ser provido a um ou mais do TCS 1378, o TESS 1380, o TES 1382, o GS 1384, o rasterizador 1386 e o SF 1390. Deste modo, um ou mais do TCS 1378, TESS 1380, o TES 1382, o GS 1384, o rasterizador 1386 e o SF 1390 podem mudar processamento (por exemplo, mudar nível de tecelagem) de um conjunto de dados de acordo com o parâmetro de LOD atualizado durante o processamento do conjunto de dados, que foi iniciado com uma ou mais diferentes parâmetros.

[00181] O circuito VLIW 1304 (por exemplo, pipeline gráfico 1306) pode incluir um controlador de buffer de quadro não uniforme 1312. Como ilustrado na Figura 13, o controlador de buffer de quadro não uniforme 1312 pode ser incluído e/ou implementado no FS 1390. Deve ser notado que o controlador de buffer de quadro não uniforme 1312 pode ser adicionalmente ou em alternativa incluído e/ou implementado em um ou mais do FS 1390, o pipeline gráfico 1306 e memória em chip 1310. Completa ou parcialmente, por exemplo, um controlador de buffer de quadro não uniforme 1312 pode ser

implementado em um ou mais do FS 1390, pipeline gráfico 1306, memória em chip 1310 e o dispositivo eletrônico 1302.

[00182] O controlador de buffer de quadro não uniforme 1312 descrito em conexão com a Figura 13, pode ser um exemplo de um ou mais dos controladores de buffer de quadro não uniformes 112, 312 descritos em ligação com uma ou mais das Figuras 1-5. Por exemplo, o controlador de buffer de quadro não uniforme 1312 pode acessar dinamicamente diferentes subconjuntos de um ou mais buffer (s) de quadro (por exemplo, buffer de quadro 1341). Em algumas configurações, dinamicamente acessar diferentes subconjuntos do buffer de quadro pode incluir o acesso a pelo menos um pixel de uma primeira zona a uma taxa diferente de pelo menos um pixel de uma segunda zona. Adicionalmente ou em alternativa, dinamicamente acessar diferentes subconjuntos do buffer de quadro pode incluir permitir troca de sub-rotina de shader dinâmico (quando o processamento de um primeiro pixel está em andamento e processo de um segundo pixel é finalizado, por exemplo). Adicionalmente ou em alternativa, o controlador de buffer de quadro não uniforme pode ser configurado para substituir uma primeira amostra processada de um primeiro pixel para uma segunda amostra de um segundo pixel para evitar o processamento da segunda amostra do segundo pixel. A arquitetura ilustrada na Figura 13 pode permitir a configuração personalizada dos buffers em chip (por exemplo, buffers de profundidade e cor). Isto pode permitir que o renderizar lado-a-lado para VR, geração de mapa de bits, etc.

[00183] A figura 14 é um diagrama de blocos que ilustra um outro exemplo de um dispositivo eletrônico 1466

que inclui uma circuito de palavra de instrução muito longa (VLIW) 1468. Em particular, a Figura 14 ilustra um exemplo de uma arquitetura de processamento gráfico. Esta arquitetura pode prover (por exemplo, suportar) tanto modo imediato (por exemplo, IMR) quanto modo em mosaico (por exemplo, TBDR). Por exemplo, a arquitetura pode prover uma seleção de modo imediato ou modo em mosaico. O exemplo ilustrado na Figura 14 pode evitar grande cache / buffers em chip (necessários para a geometria intermediária, por exemplo).

[00184] O dispositivo eletrônico 1466 pode incluir um circuito VLIW 1468 (por exemplo, GPU) e a memória de sistema 1405 (por exemplo, DRAM). O circuito VLIW 1468 pode incluir um pipeline gráfico 1470 e memória em chip 1403. O pipeline gráfico 1470 pode incluir um ou mais de um processador de vértice (VP) 1474, shader de vértice (VS) 1476, interruptores AB 1447a-b (por exemplo, multiplexadores), shader de controle de tecelagem (TCS) 1478, tessellator (TESS) 1480 (por exemplo, máquina de tecelagem), shader de avaliação de tecelagem (TES) 1482, shader de geometria (GS) 1484, ladrilhador 1449, rasterizador (RAST) 1486 e shader de fragmento (FS) 1490. Muitos dos rótulos de elementos foram abreviados por conveniência. A memória em chip 1403 pode incluir um cache 1401 e um buffer de profundidade e cor 1443. A memória de sistema 1405 pode incluir dados de geometria 1409, um buffer de parâmetro 1435, dados de textura 1411 e um buffer de quadro 1441. Os dados de geometria 1409 podem indicar um ou mais pontos (por exemplo, os vértices), as linhas e/ou

polígonos. Os dados de geometria 1409 podem representar uma cena e/ou um ou mais objetos em uma cena.

[00185] O VP 1474 pode ir buscar dados de geometria provenientes da memória de sistema 1405 para o circuito VLTW 1468 (por exemplo, pipeline gráfico 1470). O VP 1474 pode prover os dados de geometria para o VS 1476.

[00186] O VS 1476 pode transformar (por exemplo, projeto) os dados de geometria em um espaço diferente. Por exemplo, o VS 1476 pode transformar os dados de geometria do espaço tridimensional (3D) em um espaço bidimensional (2D). Em algumas configurações, o VS 1476 pode executar recorte, projeção e/ou descarte. Por exemplo, o VS 1476 pode recortar dados de geometria que estão fora do frustum de visualização. O VS 1476 pode projetar os dados de geometria do espaço 3D para o espaço 2D. O VS 1476 também pode descartar alguns dos dados de geometria. Por exemplo, os dados de geometria correspondentes à geometria ocluída (por exemplo, a geometria que não é visível porque está bloqueada por outra geometria) pode ser removida. O VS 1476 pode prover os dados de geometria transformados para o interruptor A 1447a (por exemplo, um multiplexador).

[00187] O interruptor A 1447a pode rotear os dados de geometria transformados para o TCS 1478 ou para o ladrilhador 1449 com base em um modo 1445. Por exemplo, o modo 1445 pode indicar um modo de renderização. Em algumas configurações, os modos de renderização podem incluir modo em mosaico (por exemplo, modo de renderização diferido baseado em mosaico (TBDR)) e modo imediato (por exemplo, renderização de modo imediato (IMR)). Se tecelagem está habilitada, o interruptor A 1447a pode rotear os dados de

geometria transformados para ser tecelados. Caso contrário, o interruptor A 1447a pode rotear os dados de geometria transformados para estarem em mosaico. Por exemplo, em um caso em que o modo 1445 indica o modo em mosaico, o interruptor A 1447a pode direcionar a geometria transformada para o ladrilhador 1449. Em um caso em que o modo 1445 indica o modo imediato, o interruptor A 1447a pode dirigir a geometria transformada no TCS 1478. O modo imediato pode incluir uma ou mais das operações descritas em ligação com uma ou mais das Figuras 8 e 10. O modo em mosaico pode incluir uma ou mais das operações descritas em ligação com a Figura 12.

[00188] O TCS 1478 pode receber dados de geometria (por exemplo, dados de geometria transformados) de interruptor A 1447a (por exemplo, o VS 1476). O TCS 1478 pode transformar a base dos dados de geometria (por exemplo, os dados de geometria transformados). Por exemplo, o TCS 1478 pode transformar a base dos dados de geometria para uma base para tecelagem (por exemplo, para subdividir eficientemente polígonos tal como triângulos). Por exemplo, o TCS 1478 pode produzir um indicador de base que indica a transformação de base realizada pelo TCS 1478. Os dados de geometria transformados na base podem ser provados para o TESS 1480.

[00189] O TESS 1480 pode tecelar os dados de geometria (por exemplo, dados de geometria transformados na base). Por exemplo, o TESS 1480 pode gerar mais dados de geometria (por exemplo, os vértices) com base nos dados de geometria (por exemplo, pode subdividir triângulos em mais triângulos). O TESS 1480 pode operar com base em um parâmetro

de LOD. O parâmetro de LOD pode especificar o LOD e/ou pode indicar um grau de tecelagem para o TESS 1480 executar. Os dados de geometria tecelados podem ser provados para o TES 1482.

[00190] O TES 1482 pode transformar a base dos dados de geometria (por exemplo, dados de geometria tecelados). Por exemplo, o TES 1482 pode transformar a base dos dados de geometria de volta para a base antes da transformada de base aplicada pelo TCS 1478. Em algumas configurações, o TES 1482 pode obter o indicador de base a partir do SCT 1478 e/ou memória (por exemplo, memória em chip 1403 e/ou memória de sistema 1405), a fim de transformar (por exemplo, transformada inversa) a base dos dados de geometria. O TES 1482 pode prover dados de geometria (por exemplo, os dados de geometria transformados na base inversa) para o GS 1484.

[00191] O GS 1484 pode executar uma ou mais operações sobre os dados de geometria (por exemplo, os dados de geometria transformados na base inversa). Por exemplo, o GS 1484 pode gerar geometria adicional com base nos dados de geometria. O GS 1484 pode prover os dados de geometria (com geometria adicional, por exemplo) para alternar B 1447b. Interruptor B 1447b (por exemplo, um multiplexador) pode rotear os dados de geometria (com geometria adicional, por exemplo) para o rasterizador 1486 (no caso do modo imediato, por exemplo).

[00192] Em um caso em que o modo 1445 indica o modo TBDR, o interruptor A 1447a pode prover os dados de geometria transformados para o ladrilhador 1449 (a partir do VS 1476, por exemplo). O ladrilhador 1449 pode dividir os

dados de geometria transformados em mosaicos. Por exemplo, o ladrilhador 1449 pode dividir os dados de geometria em grupos menores de dados de geometria. Os mosaicos podem incluir dados de geometria contíguos. O ladrilhador 1449 pode prover os mosaicos para o buffer de parâmetro 1435.

[00193] O buffer de parâmetro 1435 pode armazenar os mosaicos. Por exemplo, o buffer de parâmetro 1435 pode armazenar uma lista de primitivas (por exemplo, dados de geometria, vértices, linhas, polígonos, etc.). No modo em mosaico descrito na Figura 14, o processamento de mosaico pode alavancar localidade espacial de memória de acessos ao sistema. Por exemplo, operações que são locais a um mosaico podem ser relativamente "baratas" (por exemplo, podem utilizar muito pouco processamento, energia, memória e/ou recursos de tempo). Os dados de geometria (por exemplo, dados de geometria em mosaico, primitivas, etc.) podem ser providos para o interruptor B 1447b. Interruptor B 1447b pode rotear os dados de geometria para o rasterizador 1486 do buffer de parâmetro 1435 (no caso do modo em mosaico, por exemplo).

[00194] O rasterizador 1486 pode receber os dados de geometria (por exemplo, dados de geometria em mosaico a partir do buffer de parâmetro 1435 ou dados de geometria (com geometria adicional, por exemplo) a partir do GS 1484). O rasterizador 1486 pode rasterizar os dados de geometria. Por exemplo, o rasterizador pode converter os dados de geometria de dados de pixel. Os dados de pixel podem ser providos para o FS 1490.

[00195] O FS 1490 pode executar shader de fragmento ou pixel shader nos dados de pixel. Por exemplo, o FS 1490 pode determinar um ou mais atributos (por exemplo, cor,

brilho, transparência, etc.) de um ou mais pixels. Por exemplo, o FS 1490 pode executar uma ou mais das operações e/ou pode incluir uma ou mais das estruturas descritas acima em conexão com um ou mais do FSS 890, 990, 1090, 1190, 1290, 1390 descrito em conexão com uma ou mais das Figuras 8-12.

[00196] O FS 1490 pode prover dados de pixels a um ou mais do cache 1401 e buffer de profundidade e cor 1443. Em algumas configurações, o FS 1490 e/ou o buffer de profundidade e cor 1443 podem prover uma funcionalidade semelhante aos blocos de composição / decomposição aqui descritos. Em algumas configurações, a arquitetura descrita em ligação com a Figura 14 pode não misturar o modo imediato e modo em mosaico. Por exemplo, o modo (por exemplo, o modo 1445) pode indicar modo imediato de tecelagem ou modo em mosaico sem tecelagem.

[00197] O cache 1401 pode receber dados de textura 1411 da memória de sistema. Em algumas configurações, apenas acessos de textura e mosaicos de buffer de quadro podem ser armazenados em cache. No caso de modo imediato, um mosaico pode ser alocado que é o tamanho de todo o buffer de quadro 1441.

[00198] Conforme ilustrado na Figura 14, os dados do buffer de cor e de profundidade podem ser armazenados no chip. O buffer de profundidade e cor 1443 pode prover os dados de pixel para o buffer de quadro 1441. O buffer de quadro 1441 pode emitir os dados de pixel. Os dados de pixel podem ser providos (por exemplo, emitidos) para apresentação. Por exemplo, os dados de pixel podem ser enviados para um display para apresentação.

[00199] Deve ser notado que, sem os sistemas e métodos aqui divulgados, as operações descritas em ligação com a Figura 14 podem ser realizadas em termos de conjuntos de dados (por exemplo, conjuntos de vértice, dados de pixels e/ou imagem), como descrito em conexão com a Figura 8.

[00200] A figura 15 é um diagrama de blocos que ilustra um exemplo dos sistemas e métodos aqui descritos implementados na arquitetura descrita em ligação com a Figura 14. Esta arquitetura pode prover modo imediato (por exemplo, IMR) e modo em mosaico (por exemplo, TBDR). Especificamente, a Figura 15 ilustra um dispositivo eletrônico 1502. O dispositivo eletrônico 1502 descrito em conexão com a Figura 15 pode ser um exemplo de um ou mais dos dispositivos eletrônicos 102, 302 descritos acima em ligação com uma ou mais das Figuras 1-5. O dispositivo eletrônico 1502 pode ser configurado para executar um ou mais dos métodos, funções, etapas, operações, etc., descritos em ligação com uma ou mais das Figuras 1-5.

[00201] O dispositivo eletrônico 1502 pode incluir um circuito VLIW 1504 (por exemplo, uma GPU) e memória de sistema 1514. O circuito VLIW 1504 pode ser um exemplo de um ou mais dos circuitos VLIW 104, 304 descritos acima em conexão com uma ou mais das Figuras 1-5, Adicionalmente ou em alternativa, a memória de sistema 1514 pode ser um exemplo de uma ou mais das memórias de sistema 114, 314 descritas em ligação com uma ou mais das Figuras 1-5. A memória de sistema 1514 pode incluir dados de geometria 1509, um buffer de parâmetro 1535, dados de textura 1511 e/ou um buffer de quadro 1541. Um ou mais dos dados de geometria 1509, um buffer de parâmetro 1535, dados de textura 1511 e/ou um

buffer de quadro 1541 podem ser semelhantes aos elementos descritos em ligação com a Figura 14 correspondente. Deve ser notado que a memória de sistema 1514 pode incluir um ou mais buffers de quadro, como descrito em ligação com uma ou mais das Figuras 3-4. Por exemplo, o buffer de quadro 1541 pode ser um exemplo de um ou mais dos buffers de quadro descritos em ligação com uma ou mais das Figuras 3-4. Em algumas configurações, um ou mais buffers de quadro pode ser incluído na memória em chip 1510, a memória de sistema 1514 ou ambos.

[00202] O circuito VLIW 1504 pode incluir um pipeline gráfico 1506 e memória em chip 1510. O pipeline gráfico 1506 descrito em ligação com a Figura 15 pode ser um exemplo de um ou mais dos pipelines gráficos 106, 306 descritos em ligação com uma ou mais das Figuras 1-5. Adicionalmente ou em alternativa, a memória em chip 1510 descrita em conexão com a Figura 15, pode ser um exemplo de uma ou mais das memórias em chip 110, 310 descritas em ligação com uma ou mais das Figuras 1-5.

[00203] A memória em chip 1510 pode incluir um cache 1501 e/ou um buffer de profundidade e cor 1543. Em algumas configurações, um ou mais da memória em chip 1510, cache 1501 e/ou um buffer de profundidade e cor 1543 podem ser semelhantes aos elementos descritos em ligação com a Figura 14 correspondente.

[00204] O pipeline gráfico 1506 pode incluir um ou mais de um VP 1574, VS 1576, interruptor AB 1547a-b, TCS 1578, TESS 1580, TES 1582, GS 1584, ladrilhador 1549, rasterizador 1586 e FS 1590. Em algumas configurações, um ou mais do VP 1574, VS 1576, interruptor 1547a-b, TCS 1578,

TESS 1580, TES 1582, GS 1584, ladrilhador 1549, rasterizador 1586 e FS 1590 pode executar uma ou mais das operações de um ou mais elementos correspondentes descritos em ligação com a Figura 14. Deve ser notado que o FS 1590 pode incluir um ou mais de um HSR & DT, TB, T & S, AT e AB como descrito em conexão com a Figura 14, em algumas configurações.

[00205] Nas configurações ilustradas na Figura 15, o circuito VLIW 1504 (por exemplo, pipeline gráfico 1506) pode incluir um controlador de memória assíncrona 1508. O controlador de memória assíncrona 1508 pode ser incluído e/ou implementado em um ou mais do TCS 1578, o TESS 1580, o TES 1582, o GS 1584, o rasterizador 1586, o SF 1590 e o ladrilhador 1549. Completa ou parcialmente, por exemplo, um controlador de memória assíncrona 1508 pode ser implementado no TCS 1578, o TESS 1580, o TES 1582, o GS 1584, o rasterizador 1586, o SF 1590 e/ou o ladrilhador 1549. O rótulo genérico "1508" pode indicar que o controlador de memória assíncrona 1508 pode ser implementada em qualquer um ou todos os elementos 1508a-g ilustrados na Figura 15. Em algumas configurações, o controlador de memória assíncrona 1508 pode ser implementado apenas em um ou mais do TCS 1578, TESS 1580, TES 1582 e GS 1584. Por exemplo, o controlador de memória assíncrona 1508 somente pode ler de forma assíncrona um ou mais parâmetros no TCS 1578, TESS 1580, TES 1582 e/ou GS 1584, uma vez que os dados podem ser transmitidos para as fases posteriores.

[00206] O interruptor 1547a pode rotear os dados para o TCS 1578 ou para o ladrilhador 1549 com base em um modo 1545. Por exemplo, o modo 1545 pode indicar um modo de renderização. Em algumas configurações, os modos de

renderização podem incluir modo em mosaico (por exemplo modo de renderização diferido baseado em mosaico (TBDR)) e o modo imediato (por exemplo, renderização de modo imediato (EVIR)). Se tecelagem está habilitada, o interruptor A 1547a pode encaminhar os dados a serem tecelados. Caso contrário, o interruptor A 1547a pode rotear os dados para estar em mosaico. Modo imediato pode incluir uma ou mais das operações descritas em ligação com uma ou mais das Figuras 8 e 10. Modo em mosaico pode incluir uma ou mais das operações descritas em ligação com a Figura 12.

[00207] No modo em mosaico, o controlador de memória assíncrona 1508e-g incluído em um ou mais do ladrilhador 1549, rasterizador 1586 e FS 1590 pode executar zoneamento e/ou reduzir o cálculo do FS 1590 em algumas configurações. Por exemplo, o controlador de memória assíncrona 1508e-g pode empregar um shader mais oneroso para uma zona de alto LOD, um shader moderadamente oneroso para uma zona de LOD médio e um shader mais barato para uma zona de LOD baixo. Por exemplo, renderização foveada pode incluir o LOD de geometria e/ou cálculo de shader. Em algumas configurações (no modo em mosaico, por exemplo), a geometria não pode ser modificada, mas um ou mais shaders que são executados por pixel podem ter diferentes despesas de computação.

[00208] O controlador de memória assíncrona 1508 descrito em conexão com a Figura 15, pode ser um exemplo de um ou mais dos controladores de memória assíncrona 108, 308 descritos em ligação com uma ou mais das Figuras 1-5. Por exemplo, o controlador de memória assíncrona 1508 pode de forma assíncrona ler um ou mais parâmetros (por exemplo,

parâmetros de LOD) a partir da memória de sistema 1514 (durante o processamento de um conjunto de dados, por exemplo). Em algumas configurações, o parâmetro (s) atualizado pode ser provido a um ou mais do TCS 1578, o TESS 1580, o TES 1582, o GS 1584, o rasterizador 1586, o SF 1590 e o ladrilhador 1549. Por conseguinte, um ou mais do TCS 1578, o TESS 1580, o TES 1582, o GS 1584, o rasterizador 1586, o FS 1590 e o ladrilhador 1549 podem alterar o processamento do conjunto de dados para ser diferente do processamento inicialmente estabelecido para um conjunto de dados, enquanto o conjunto de dados está sendo processado.

[00209] Em algumas configurações, isto pode permitir que o dispositivo eletrônico 1502 realize renderização de única passagem multi-LOD. Por exemplo, o controlador de memória assíncrona 1508 pode ler um parâmetro de LOD que é diferente de um outro parâmetro de LOD inicialmente estabelecido para o conjunto de dados que está atualmente sendo processado e/ou renderizado (por exemplo, tecelado). O parâmetro de LOD atualizado pode ser provido a um ou mais do TCS 1578, o TESS 1580, o TES 1582, o GS 1584, o rasterizador 1586, o SF 1590 e o ladrilhador 1549. Deste modo, um ou mais do TCS 1578, o TESS 1580, o TES 1582, o GS 1584, o rasterizador 1586, o SF 1590 e o ladrilhador 1549 podem mudar de renderização (por exemplo, alterar o nível de mosaico e/ou alterar um ou mais shaders) de um conjunto de dados de acordo com o parâmetro de LOD atualizado durante o processamento do conjunto de dados que foi iniciado com um ou mais parâmetros diferentes.

[00210] O circuito VLIW 1504 (por exemplo, pipeline gráfico 1506) pode incluir um controlador de buffer

de quadro não uniforme 1512. Como ilustrado na Figura 15, o controlador de buffer de quadro não uniforme 1512 pode ser incluído e/ou implementado no FS 1590. Deve ser notado que o controlador de buffer de quadro não uniforme 1512 pode ser adicionalmente ou em alternativa incluído e/ou implementado em um ou mais do FS 1590, pipeline gráfico 1506 e memória em chip 1510. Completa ou parcialmente, por exemplo, uma controlador de buffer de quadro não uniforme 1512 pode ser implementado em um ou mais do FS 1590, pipeline gráfico 1506, memória em chip 1510 e o dispositivo eletrônico 1502.

[00211] O controlador de buffer de quadro não uniforme 1512 descrito em conexão com a Figura 15 pode ser um exemplo de um ou mais dos controladores de buffer de quadro não uniformes 112, 312 descritos em ligação com uma ou mais das Figuras 1-5. Por exemplo, o controlador de buffer de quadro não uniforme 1512 pode acessar dinamicamente diferentes subconjuntos de um ou mais buffer (s) de quadro (por exemplo, buffer de quadro 1541). Em algumas configurações, acessar dinamicamente diferentes subconjuntos do buffer de quadro pode incluir o acesso a pelo menos um pixel de uma primeira zona a uma taxa diferente de pelo menos um pixel de uma segunda zona. Adicionalmente ou em alternativa, acessar dinamicamente diferentes subconjuntos do buffer de quadro pode incluir permitir troca de sub-rotina de shader dinâmico (quando o processamento de um primeiro pixel está em andamento e processo de um segundo pixel é finalizado, por exemplo). Adicionalmente ou em alternativa, o controlador de buffer de quadro não uniforme 1512 pode ser configurado para substituir uma primeira amostra processada de um primeiro pixel para uma segunda amostra de um segundo

pixel para evitar o processamento da segunda amostra do segundo pixel. A arquitetura ilustrada na Figura 15 pode permitir a configuração personalizada dos buffers em chip (por exemplo, buffers de profundidade e cor). Isto pode permitir que renderização em mosaico de VR, geração de mapa de bits, etc.

[00212] Adicionalmente ou em alternativa, o controlador de buffer de quadro não uniforme 312 pode prover particionamento, leitura e/ou gravação de buffer de quadro eficiente. Por exemplo, o controlador de buffer de quadro não uniforme 312 pode permitir que um shader de fragmento (FS) e/ou bloco de composição / decomposição acesse o buffer de quadro de uma maneira não uniforme (por exemplo, ao gravar diferentes subgrupos do buffer de quadro com diferentes níveis de detalhe concorrentemente, gravar diferentes subconjuntos com diferentes números de amostras, etc.).

[00213] A figura 16 é um diagrama de blocos que ilustra uma configuração de um dispositivo eletrônico 1651 em que sistemas e métodos para renderizar várias zonas podem ser implementados. Exemplos do dispositivo eletrônico 1651 incluem smartphones, computadores (por exemplo, desktops, laptops, etc.), displays head-mounted (por exemplo, fones de ouvido de Realidade Virtual (RV), fones de ouvido de Realidade Aumentada (AR), etc.), dispositivos VR, dispositivos AR, tablets, reprodutores de mídia, televisores, automóveis, veículos, máquinas fotográficas, câmeras de vídeo, câmeras digitais, telefones celulares, câmeras pessoais, câmeras de ação, aeronaves, aviões, veículos aéreos não tripulados (UAVs), equipamentos de saúde, consoles de jogos, assistentes pessoais digitais

(PDAs), set-top boxes, reprodutores de mídia, etc. Em algumas configurações, o dispositivo eletrônico 1651 pode ser pelo menos parte de um módulo removível em um veículo. Por exemplo, o dispositivo eletrônico 1651 pode ser incluído em e/ou acoplado a um veículo e pode ser removível do veículo. O dispositivo eletrônico 1651 descrito em conexão com a Figura 16 pode ou não ser um exemplo de um ou mais dos dispositivos eletrônicos 102, 302, 866, 902, 1066, 1102, 1266, 1302, 1466, 1502 descritos em ligação com uma ou mais das Figuras 1-5 e 8-15. Em algumas configurações, um ou mais dos componentes ou elementos do dispositivo eletrônico 1651 podem ou não ser semelhantes aos componentes ou elementos descritos em ligação com uma ou mais das Figuras 1-5 e 8-15 correspondentes.

[00214] Em algumas configurações, o dispositivo eletrônico 1651 pode incluir uma memória 1667, um ou mais displays 1673, um ou mais sensores de imagem 1653, um ou mais sistemas ópticos 1655, uma ou mais interfaces de comunicações 1657, uma ou mais antenas 1659 e/ou um ou mais processadores 1661. Dois ou mais dos elementos ou componentes descritos em ligação com a Figura 16 podem estar em comunicação eletrônica com o outro. Por exemplo, o processador 1661 pode ser acoplado a (por exemplo, em comunicação eletrônica com) a memória 1667. O processador (s) 1661 pode incluir uma ou mais unidades de processamento central (CPUs), circuitos de palavra de instrução longa (VLIW), unidades de processamento gráfico, circuitos integrados de aplicação específica (ASIC), etc., ou qualquer combinação dos mesmos. Por exemplo, o processador 1661 pode ser um circuito VLIW 104, 304, 868, 904, 1068, 1104, 1268,

1304, 1468, 1504 descrito em ligação com uma ou mais das Figuras 1-5 e 8-15. Em algumas configurações, o processador 1661 pode ser um processador com um circuito VLIW integrado (por exemplo, GPU) ou pode ser um processador central e um circuito VLIW separado (por exemplo, GPU). Por exemplo, o processador 1661 pode ser um dispositivo de renderização de imagens. Em algumas configurações, a imagem (s) resultante pode ser apresentada em um ou mais displays 1673.

[00215] A interface de comunicação 1657 pode permitir que o dispositivo eletrônico 1651 se comunique com um ou mais outros dispositivos eletrônicos. Por exemplo, a interface de comunicação 1657 pode prover uma interface para comunicações com e/ou sem fio. Em algumas configurações, a interface de comunicação 1657 pode ser acoplada a uma ou mais antenas 1659 para a transmissão e/ou recepção de sinais de radiofrequência (RF). Adicionalmente ou em alternativa, a interface de comunicação 1657 pode permitir um ou mais tipos de comunicação de telefonia fixa (por exemplo, Barramento Serial Universal (USB), Ethernet, Video Graphics Array (VGA), Digital Visual Interface (DVI), High Definition Multimedia Interface (HDMI), DisplayPort, mobile High-Definition link (MHL), etc.).

[00216] Em algumas configurações, várias interfaces de comunicações 1657 podem ser implementadas e/ou utilizadas. Por exemplo, uma interface de comunicação 1657 pode ser uma interface de comunicação celular (por exemplo, 3G, Evolução de Longo Prazo (LTE), CDMA, etc.) 1657, outra interface de comunicação 1657 pode ser uma interface Ethernet celular, uma outra interface de comunicação 1657 pode ser uma interface de barramento universal serial (USB), ainda

uma interface de comunicação 1657 pode ser uma interface de rede de área local sem fio (WLAN) (por exemplo, a interface do Institute of Electrical and Electronics Engineers (IEEE) 802.11), e ainda outra interface de comunicação 1657 pode ser uma interface Bluetooth.

[00217] Em algumas configurações, o dispositivo eletrônico 1651 pode executar uma ou mais das funções, processos, métodos, etapas, etc., descritas em ligação com uma ou mais das Figuras 1-15 e 17-24. Adicionalmente ou em alternativa, o dispositivo eletrônico 1651 pode incluir uma ou mais das estruturas descritas em ligação com uma ou mais das Figuras 1-15 e 17-24.

[00218] A memória 1667 pode armazenar instruções e/ou dados. A memória 1667 pode incluir a memória de sistema e/ou memória em chip. A memória 1667 descrita em conexão com a Figura 16 pode ou não ser um exemplo de uma ou mais das memórias 110, 114, 310, 314, 803, 805, 910, 914, 1003, 1005, 1110, 1114, 1203, 1205, 1310, 1314, 1403, 1405, 1510, 1514 descritas em ligação com uma ou mais das Figuras 1-5 e 8-15. O processador 1661 pode acessar (por exemplo, ler a partir de e/ou gravar) a memória 1667. A memória 1667 pode armazenar dados de geometria 1669. Em algumas configurações, a memória 1667 pode incluir um buffer de quadro 1671. O buffer de quadro 1671 pode ser uma porção da memória 1667 para armazenar dados de imagem. Os dados de imagem armazenados em buffer podem ser apresentados em uma ou mais displays 1673 em algumas configurações.

[00219] Em algumas configurações, o processador 1661 pode incluir um ou mais elementos de renderização de imagens. Alguns exemplos dos elementos pode incluir um front

end (EF), processador de vértice (VPS, coletor de vértice (VF), shader de vértice (VS), shader de controle de tecelagem (TCS), hull shader (HS), tessellator (por exemplo, máquina de tecelagem) (TESS), shader de avaliação de tecelagem (TES), shader de domínio (DS), shader de shader (GS), rasterizador (RAST), shader de fragmento (FS), pixel shader (PS), ladrilhador, unidade de saída de renderização (ROP), clipper (clip) e/ou saída de fluxo (SOL), etc. Em algumas configurações, um ou mais destes elementos podem ser referidos como um pipeline gráfico.

[00220] O processador 1661 pode incluir e/ou implementar um renderizador multizona 1663. Por exemplo, o renderizador multizona pode ser implementado em hardware ou em uma combinação de hardware e software. O renderizador multizona 1663 pode renderizar várias zonas de uma imagem. Por exemplo, o renderizador multizona 1663 pode processar várias zonas de uma imagem com base nos dados de geometria 1669. As várias zonas podem ser fundidas com diferentes níveis de detalhe tecelados. Por exemplo, o renderizador multizona 1663 pode renderizar N zonas, onde pelo menos duas das zonas têm diferentes níveis tecelados de detalhe. Por exemplo, o renderizador multizona pode renderizar uma primeira zona de uma imagem a um primeiro nível tecelado de detalhe e pode renderizar uma segunda zona da imagem com um segundo nível tecelado de detalhe mais elevado. As zonas podem ser estáticas ou dinâmicas. Por exemplo, as zonas podem estar em locais predeterminados em um display 1673. Por conseguinte, o renderizador multizona 1663 pode renderizar a geometria ao nível de detalhe que corresponde às zonas estáticas. Em outras configurações, as zonas podem ser

dinâmicas. Por exemplo, as zonas podem mover-se com base no rastreamento ocular e/ou outros fatores (por exemplo, tipo de objeto, prioridade, definições recebidas, etc.). Em algumas configurações, a zona pode ser uma área circular (ou a uma região de outra forma) aproximadamente centrada no ponto de um display (por exemplo, imagem) onde o usuário está olhando. Em outras configurações, uma zona pode ser toda a área de uma imagem que corresponde a uma janela que o usuário está visualizando.

[00221] O dispositivo eletrônico 1651 pode ser acoplado a um ou mais displays 1673. Por exemplo, o dispositivo eletrônico 1651 pode incluir um ou mais displays 1673 e/ou pode ser acoplado a um ou mais displays externos (por exemplo, remoto) 1673. Exemplos do display 1673 incluem uma tela de cristal líquido (LCD), painéis de diodo emissor de luz (LED) (por exemplo, painéis de diodos emissores de luz orgânica (OLED)), displays de tubo de raios catódicos (CRT), monitores, televisões, projetores tela de plasma, display montado na cabeça (s), displays de tinta eletrônicos, etc.

[00222] Em algumas configurações, o display (s) 1673 pode ser capaz de apresentar imagens em uma ou mais janelas (por exemplo, janelas de veículos, janelas de construção, etc.). Por exemplo, o display (s) 1673 pode incluir um ou mais projetores de janela e/ou um ou mais painéis de exibição de janela integrados. O renderizador multizona 1663 pode processar as várias zonas (onde duas ou mais zonas podem ter diferentes níveis de detalhe, por exemplo). Em algumas configurações, o dispositivo eletrônico 1651 (por exemplo, um processador 1661) pode apresentar zonas

diferentes em diferentes janelas. Por exemplo, uma primeira zona pode ser apresentada em um primeiro indicador (por exemplo, para-brisas), uma segunda zona pode ser apresentada em uma segunda janela (por exemplo, a janela de porta de veículo (s)) e/ou uma terceira zona pode ser apresentada em uma terceira janela (por exemplo, janela de trás do veículo). Em algumas configurações, diferentes zonas podem ser apresentadas em seções diferentes da mesma janela. Por exemplo, uma primeira zona pode ser apresentada em uma primeira seção de uma janela (por exemplo, uma seção lateral de um para-brisa do condutor) e uma segunda zona pode ser apresentada em uma segunda seção da janela (por exemplo, uma seção lateral do para-brisa do passageiro). Em outro exemplo, as zonas podem ser apresentadas como barras (por exemplo, barras horizontais ou verticais) sobre uma janela. Por exemplo, uma primeira zona pode ser apresentada em uma barra horizontal no meio de uma janela (por exemplo, para-brisas), enquanto uma segunda zona pode ser apresentada em barras horizontais inferiores e superiores da janela (por exemplo, para-brisas). Em algumas configurações, uma combinação das abordagens anteriores pode ser utilizada. Por exemplo, uma primeira zona pode ser apresentada em um lado do condutor de um para-brisa, uma segunda zona pode ser provida em um lado do passageiro do para-brisa e uma terceira zona pode ser apresentada em janelas da porta do veículo. Outras variações podem ser implementadas.

[00223] Em algumas configurações, o dispositivo eletrônico 1651 pode incluir um controlador de memória assíncrona e/ou um controlador de buffer de quadro não uniforme. Por exemplo, o renderizador multizona 1663 pode

incluir e/ou ser acoplado a um controlador de memória assíncrona e/ou a um controlador de buffer de quadro não uniforme. O renderizador multizona 1663 pode ou não renderizar as várias zonas de acordo com as operações, funções e/ou abordagens descritas em ligação com o controlador de memória assíncrona e/ou com o controlador de buffer de quadro não uniforme, tal como aqui descrito. Por exemplo, o dispositivo eletrônico 1651 pode ou não realizar renderização multi-LOD de passagem única como aqui descrito.

[00224] O dispositivo eletrônico 1651 pode obter uma ou mais imagens (por exemplo, imagens digitais, quadros de imagens, vídeo, etc.). As uma ou mais imagens podem ser imagens de um ou mais usuários e/ou uma cena (por exemplo, um ou mais objetos e/ou fundo). Por exemplo, o dispositivo eletrônico 1651 pode incluir um ou mais sensores de imagem 1653 e/ou um ou mais sistemas ópticos 1655 (por exemplo, lentes) que focam imagens de objetos que se encontram dentro do campo de visualização do sistema óptico (s) 1655 sobre o sensor de imagem (s) 1653. Em algumas configurações, o sensor (s) de imagem 1653 pode captar uma ou mais imagens. O sensor (s) de imagem 1653 e/ou o sistema óptico (s) 1655 podem ser acoplados a e/ou controlados pelo processador 1661.

[00225] Adicionalmente ou em alternativa, o dispositivo eletrônico 1651 pode solicitar e/ou receber uma ou mais imagens a partir de outro dispositivo (por exemplo, um sensor de imagem externo acoplado ao dispositivo eletrônico 1651, um servidor de rede, câmera de tráfego, câmera drop, câmera de automóvel, câmera web, etc.). Em algumas configurações, o dispositivo eletrônico 1651 pode solicitar e/ou receber uma ou mais imagens através da

interface de comunicação 1657. Por exemplo, o dispositivo eletrônico 1651 pode ou não incluir uma câmara (por exemplo, um sensor de imagem 1653 e/ou sistema óptico 1655) e pode receber imagens de um dispositivo remoto.

[00226] Em algumas configurações, os um ou mais sensores de imagem 1653 incluídos no dispositivo eletrônico 1651 e/ou um ou mais outros sensores de imagem acoplados ao dispositivo eletrônico 1651 podem ser orientados para captar uma ou mais imagens de um usuário. Por exemplo, um aparelho pode incluir uma câmara montada na mesma face que um display (por exemplo, display 1673). Desta forma, a câmera pode capturar imagens de um usuário enquanto o usuário está olhando para o display. Em um outro exemplo, uma câmara pode ser montada em um console e/ou volante de veículo para capturar imagens de um usuário (por exemplo, condutor).

[00227] Adicionalmente ou em alternativa, os um ou mais sensores de imagem 1653 incluídos no dispositivo eletrônico 1651 e/ou um ou mais outros sensores de imagem acoplados ao dispositivo eletrônico 1651 podem ser orientados para capturar cenário. Por exemplo, uma ou mais câmeras podem ser montadas na parte de trás de um smartphone. Em um outro exemplo, uma ou mais câmaras podem ser montadas no exterior de um veículo. Em algumas configurações, o dispositivo eletrônico 1651 pode utilizar as imagens do cenário em aplicações de realidade aumentada (AR), aplicações de realidade virtual (VR) e/ou outras aplicações (por exemplo, visualização de vista surround, assistência ao condutor, sistemas de assistência ao condutor avançados (ADAS), etc.).

[00228] Em algumas configurações, a imagem (s) capturada pelo sensor (s) de imagem 1653 e/ou recebida através da interface (s) de comunicação 1657 pode ser utilizada pelo dispositivo eletrônico 1651 para o rastreamento ocular. Por exemplo, o processador 1661 pode incluir e/ou implementar um rastreamento ocular 1665. O rastreamento ocular 1665 pode controlar o olhar (por exemplo, direção do olho) do olho (s) de um ou mais usuários. Por exemplo, o rastreamento ocular 1665 pode localizar o olho (s) de um ou mais usuários na imagem (s) e pode estimar o olhar (por exemplo, em direção do olho) do usuário (es). Por exemplo, o rastreamento ocular 1665 pode determinar (por exemplo, estimar) onde um usuário está olhando em um display (por exemplo, display 1673 ou um display externo) e/ou onde um usuário está olhando em uma imagem apresentada por um display. Em outras palavras, o rastreamento ocular 1665 pode determinar (por exemplo, estimar) para onde olhar de um usuário está sendo direcionado em relação a uma imagem e/ou a um display. Em algumas configurações, o rastreamento ocular 1665 pode realizar rastreamento ocular de forma semelhante ao rastreamento ocular 326 descrito acima em ligação com a Figura 3.

[00229] Em algumas configurações, o renderizador multizona 1663 pode renderizar uma ou mais zonas de uma imagem com base no processamento ocular. As zonas da imagem podem ter diferentes níveis de detalhe. Por exemplo, o renderizador multizona 1663 pode renderizar múltiplas zonas com diferentes níveis de detalhe, onde uma zona com um nível de detalhe mais alto é renderizada em uma área onde um usuário está atualmente olhando. Em algumas configurações,

isso pode ser chamado de renderização foveada. O rastreador ocular 1665 pode produzir informação de rastreamento ocular. A informação de rastreamento ocular pode indicar para onde o olhar de um usuário é direcionado (por exemplo, os olhares dos usuários são direcionados) em relação a uma exibição e/ou imagem. Em algumas configurações, o processador 1661 e/ou o processador 1661 podem determinar um ou mais parâmetros de LOD baseados na informação de rastreamento ocular. O (s) parâmetro (s) de LOD pode indicar um LOD a ser renderizado para uma ou mais zonas de uma imagem. Os parâmetros de LOD podem ser armazenados na memória 1667. Os parâmetros de LOD podem ou não ser atualizados de forma assíncrona em relação a um conjunto de dados (por exemplo, chamar de desenho) que está sendo processado pelo processador 1661.

[00230] O rastreamento ocular 1665 pode detectar quando um usuário muda o olhar para olhar para uma parte diferente da imagem. Os sistemas e métodos aqui divulgados podem alterar a renderização de LOD com base no rastreamento ocular (por exemplo, a informação de rastreamento ocular e/ou parâmetro (s) de LOD). Por exemplo, o renderizador multizona 1663 pode aumentar o nível de detalhe em uma área da imagem para a qual o usuário alterou o olhar para visualização. Adicionalmente ou em alternativa, os sistemas e métodos aqui descritos podem reduzir o LOD em uma área da imagem para a qual o olhar do usuário já não está direcionado. Como descrito acima, diferentes zonas podem corresponder a diferentes janelas em algumas configurações. O renderizador multizona 1663 pode mudar a zona que corresponde a uma janela (e/ou uma seção de janela), com

base em qual janela o usuário está olhando. Por exemplo, o renderizador multizona 1663 pode renderizar uma zona de LOD elevado no para-brisa de um veículo, enquanto o usuário (por exemplo, condutor) está olhando para (por exemplo, através) do para-brisa. Quando um usuário (por exemplo, condutor) olha para outra janela (por exemplo, volta-se para verificar um ponto cego e/ou para dirigir em sentido inverso, etc.), o renderizador multizona 1663 pode mudar a zona de LOD elevado para a janela (por exemplo, janela lateral e/ou janela de trás) para onde o usuário está olhando (por exemplo, através de).

[00231] Em algumas configurações, o rastreamento ocular 1665 pode controlar um ou mais olhos de cada um dos vários usuários (por exemplo, os olhos de um condutor e um ou mais olhos dos passageiros). Os sistemas e métodos aqui descritos podem dar prioridade a renderização de zona (por exemplo, renderização de LOD) com base no rastreamento ocular (por exemplo, a informação de rastreamento ocular e/ou parâmetro (s) de LOD). Em algumas abordagens, o renderizador multizona 1663 pode prover um LOD superior para uma zona correspondente a um usuário (ou para zonas correspondentes a um conjunto de usuários de prioridade, por exemplo) e um ou mais LODs inferiores para uma ou mais zonas correspondentes a um ou mais outros usuários. Adicionalmente ou em alternativa, o renderizador multizona 1663 pode renderizar uma primeira zona correspondente a um primeiro usuário e pode renderizar uma ou mais outras zonas posteriormente (em uma sequência, por exemplo) correspondentes a um ou mais outros usuários. Por exemplo, o renderizador multizona 1663 pode prover maior LOD em uma

área da imagem a ser visualizada por um controlador e/ou pode renderizar a imagem antes em uma área da imagem sendo vista pelo condutor. O renderizador multizona 1663 pode prover LOD inferior em uma área da imagem sendo vista por um passageiro e/ou podem renderizar a imagem de uma área da imagem sendo vista pelo passageiro que depois daquela do condutor. Em algumas configurações, a prioridade pode ser predeterminada (por exemplo, um fabricante e/ou o usuário pode definir uma prioridade), pode ser com base na localização do usuário (por exemplo, o condutor do veículo fica com a prioridade mais alta e o passageiro (s) fica com a prioridade mais baixa) e/ou pode ser com base no reconhecimento do usuário (por exemplo, um usuário reconhecido fica com a prioridade mais alta, outros usuários reconhecidos e/ou não reconhecidos ficam com a prioridade ou prioridades mais baixa).

[00232] Os sistemas e métodos aqui descritos podem prover melhorias na tecnologia de exibição de janela. Por exemplo, um veículo pode estar equipado com um ou mais displays de janelas (por exemplo, painel de projetor de janela (s) e/ou de exibição de janela (s)). O display (s) de janela pode ser configurado para operar em uma ou mais aplicações. Por exemplo, um display de janela de veículo pode apresentar direções de condução a um condutor. Por exemplo, o dispositivo eletrônico 1651 pode processar uma imagem com uma seta que mostra a próxima curva em uma rua lateral. Outras imagens podem ser processadas e/ou apresentadas. Por exemplo, o dispositivo eletrônico 1651 pode renderizar indicador (s) e/ou realce (s) que mostram e/ou enfatizam sinais de rua, sinais de parada, semáforos,

pedestres, informações de navegação (por exemplo, direções de condução, mapas, localização de veículos, destino, etc.), obstruções, avisos, outras informações de segurança (por exemplo, distância para outro veículo da frente, se há suficiente distância de parada na velocidade atual, etc.), informações do veículo (por exemplo, velocidade atual, indicador de combustível, rotações por minuto (RPMs), distância percorrida, temperatura, tempo, indicador de mudança de óleo, indicador de verificação de motor, temperatura do motor, aviso de temperatura), informação de entretenimento (por exemplo, estação de rádio, nome da faixa musical, duração da música, tempo de música tocada, nome do filme, duração do filme, tempo de filme reproduzido, conteúdo de filme, conteúdo de foto, etc.), informações de comunicação (por exemplo, informações de chamada telefônica, status de rede de área local (LAN), conectividade Bluetooth, etc.), etc.

[00233] Conforme o conteúdo da imagem continua a aumentar em termos de qualidade (por exemplo, geometria mais detalhada, texturas de resolução mais elevada, maiores áreas de exposição, etc.), exibição de conteúdo altamente detalhado pode ser difícil. Isto pode ser particularmente verdadeiro para as plataformas nas quais o processamento de largura de banda e/ou limitações de eficiência de potência são preocupações. Renderização de diferentes zonas, com diferentes níveis de detalhe (por exemplo, os níveis de tecelados de detalhe) pode ajudar a priorizar os recursos de renderização. Por consequência, mais detalhe pode ser provido para informação que é mais importante e/ou que é de interesse para um usuário. Por exemplo, uma zona com um LOD

superior pode ser renderizada onde um usuário está olhando e/ou é provável de olhar (por exemplo, em uma porção do meio de um para-brisa, onde os olhos do usuário estão olhando, onde um aviso está sendo exibido, etc.). Isto pode prover uma experiência aprimorada, como informações tão interessantes e/ou importantes são apresentadas com maior detalhe. Adicionalmente ou em alternativa, as áreas onde um usuário pode não perceber detalhes menores (por exemplo, na visualização periférica) e/ou zonas que tipicamente contêm as informações menos importantes (por exemplo, uma janela lateral) podem ser processadas com menos detalhe para economizar recursos de processamento e/ou energia.

[00234] Note-se que o nível tecelado de detalhe é diferente da resolução de pixels. Por exemplo, resolução de pixel pode se referir à densidade de pixels e/ou quantos pixels são dedicados a uma determinada imagem (ou área de uma imagem). Nível tecelado de detalhe pode referir-se como a geometria finamente renderizada é tecelada (por exemplo, subdividida). Por exemplo, o maior número de pontos, polígonos e/ou linhas dedicados para processar a geometria e/ou a área de uma imagem pode refletir o nível tecelado de detalhe (LOD). Adicionalmente ou em alternativa, os diferentes níveis de detalhe podem corresponder a shaders de qualidade diferentes (por exemplo, um LOD maior pode corresponder a um shader de qualidade maior, em que o processamento mais detalhado e/ou caro é realizado). Deve ser notado que em algumas configurações, a resolução de pixel pode restringir o nível de tecelagem. Por exemplo, 1 pixel pode não precisar de 100 triângulos; um mosaico 10x10 pode não precisar de 1000 triângulos, etc.

[00235] Em algumas configurações, o dispositivo eletrônico 1651 pode incluir uma aplicação de câmara e/ou um display 1673. Quando a aplicação da câmara está em execução, as imagens de objetos que se encontram dentro do campo de visualização do sistema óptico 1655 podem ser capturadas pelo sensor de imagem 1653. As imagens que estão a ser capturadas pelo sensor de imagem 1653 podem ser apresentadas no display 1673. Em algumas configurações, tais imagens podem ser apresentadas em sucessão rápida a uma taxa de quadro relativamente elevada de modo que, em qualquer dado momento, os objetos que estão localizados dentro do campo de visualização do sistema óptico 1655 são apresentados no display 1673.

[00236] Deve ser notado que um ou mais dos elementos ilustrados no dispositivo eletrônico 1651 podem ser opcionais. Por exemplo, um ou mais dos elementos ilustrados no dispositivo eletrônico 1651 podem ou não ser incluídos e/ou implementados. Adicionalmente ou em alternativa, um ou mais dos elementos ilustrados na processador 1661 pode ser implementado separadamente do processador 1661 (por exemplo, em outros circuitos, em outro processador, em um dispositivo eletrônico separado, etc.). Por exemplo, o sensor de imagem 1653, o sistema óptico 1655, a interface de comunicação 1657, o rastreamento ocular 1665 e/ou o display 1673 não podem ser implementados no dispositivo eletrônico 1651. Em algumas configurações, o dispositivo eletrônico 1651 pode receber imagens e/ou informações de rastreamento ocular de outro dispositivo. Adicionalmente ou em alternativa, um ou mais dispositivos podem ser implementados em um dispositivo separado.

[00237] Em algumas configurações, o dispositivo eletrônico 1651 pode apresentar uma interface de usuário 1675 no display 1673. Por exemplo, a interface de usuário 1675 pode permitir a um usuário interagir com o dispositivo eletrônico 1651. Em algumas configurações, o display 1673 pode ser um tela de toque que recebe a entrada a partir de toque físico (por um dedo, caneta ou outra ferramenta, por exemplo). Adicionalmente ou em alternativa, o dispositivo eletrônico 1651 pode incluir ou ser associado a uma outra interface de entrada. Por exemplo, o dispositivo eletrônico 1651 pode incluir uma câmara de frente para um usuário e pode detectar gestos de usuário (por exemplo, gestos de mão, gestos de braço, rastreamento ocular, piscadas de olho, etc.). Em outro exemplo, o dispositivo eletrônico 1651 pode ser acoplado a um mouse e pode detectar um clique do mouse. Deve ser notado que nenhuma entrada de usuário pode ser necessária em algumas configurações.

[00238] Em algumas configurações, o dispositivo eletrônico 1651 pode, opcionalmente, incluir um sensor de movimento (por exemplo, giroscópio (s), acelerômetro (s), etc.). Por exemplo, o sensor de movimento pode detectar movimento e/ou uma orientação (do dispositivo eletrônico 1651, por exemplo). Em algumas configurações, a imagem que está sendo renderizada pode ser baseada em informações de movimento e/ou de orientação. Por exemplo, o dispositivo eletrônico 1651 pode ser um display montado na cabeça (por exemplo, um aparelho telefônico de realidade virtual (RV), um aparelho telefônico de realidade aumentada (AR), etc.), um veículo (por exemplo, automóvel, drone, avião, etc.), um dispositivo móvel (por exemplo, smartphone) ou outro

dispositivo que possa utilizar informações de movimento e/ou orientação para renderizar uma imagem. Por exemplo, a informação de movimento e/ou orientação pode indicar uma direção de visualização geral em uma cena renderizada. Por exemplo, a informação de movimento e/ou orientação pode indicar uma direção de visualização geral em uma cena renderizada. O dispositivo eletrônico 1651 pode renderizar a imagem com base nesta direção de visualização em adição ou alternativamente a partir da informação de rastreamento ocular em algumas configurações.

[00239] A figura 17 é um diagrama de fluxo que ilustra um método 1700 para renderizar múltiplas zonas. O método 1700 pode ser executado pelo dispositivo eletrônico 1651 descrito em conexão com a Figura 16.

[00240] O dispositivo eletrônico 1651 pode renderizar 1702 uma primeira zona de uma imagem. Isto pode ser conseguido tal como descrito acima em ligação com a Figura 16. Por exemplo, o dispositivo eletrônico 1651 pode executar um ou mais de busca de geometria (por exemplo, busca de vértice), shader de vértice, shader de controle de tecelagem, tecelagem, shader de avaliação de tecelagem, shader de geometria, mosaicos, rasterização e/ou shader de fragmento correspondente a uma primeira zona de uma imagem.

[00241] O dispositivo eletrônico 1651 pode renderizar 1704 uma segunda zona da imagem. Isto pode ser conseguido tal como descrito acima em ligação com a Figura 16. Por exemplo, o dispositivo eletrônico 1651 pode executar um ou mais de busca de geometria (por exemplo, busca de vértice), shader de vértice, shader de controle de tecelagem, tecelagem, shader de avaliação de tecelagem, shader de

geometria, tecelagem, rasterização e/ou shader de fragmento correspondente a uma segunda zona da imagem. A primeira zona pode ter um nível tecelado mais elevado de detalhe do que a segunda zona. Por exemplo, a primeira zona pode ter uma configuração de tecelagem mais elevada (por exemplo, uma densidade de geometria mais elevada) do que uma configuração de tecelagem para a segunda zona.

[00242] O dispositivo eletrônico 1651 pode apresentar 1706 a primeira zona e a segunda zona em pelo menos uma janela de veículo. Isto pode ser conseguido tal como descrito acima em ligação com a Figura 16. Por exemplo, a primeira zona e a segunda zona podem ser apresentadas em seções separadas de uma janela (por exemplo, para-brisas) e/ou em diferentes janelas.

[00243] A figura 18 é um diagrama que ilustra alguns exemplos 1877a-b de displays de janelas. Especificamente, o exemplo A 1877a ilustra janela A 1879a (por exemplo, um para-brisa). No exemplo A 1877a, um projetor 1881 está montado de modo a projetar uma imagem na janela A 1879a. Por exemplo, um ou mais projetores 1881 podem ser montados em um painel, porta, no teto e/ou nas paredes (por exemplo, enquadramento) de um veículo. Em algumas configurações, um único projetor 1881 pode ser configurado para projetar imagens para uma ou mais janelas. Em outras configurações, vários projetores 1881 podem ser configurados para projetar imagens para uma ou mais janelas. Em algumas configurações, o projetor 1881 pode ser montado em um dispositivo removível e/ou um dispositivo móvel.

[00244] Exemplo B 1877b ilustra janela B 1879b (por exemplo, um para-brisa). No exemplo B 1877b, uma tela

integrada 1883 pode ser ligada a janela B 1879. Por exemplo, a tela integrada 1883 pode ser transparente ou semitransparente. O display integrado 1883 pode apresentar imagens na janela B 1879b. Por exemplo, a tela integrada 1883 pode ser um painel de LCD, um painel de OLED, etc. Em algumas configurações, um único projetor de exibição integrado 1883 pode ser configurado para apresentar as imagens em um ou mais janelas. Em outras configurações, várias telas integradas 1883 podem ser configuradas para apresentar as imagens em uma ou mais janelas. Deve ser notado que os projetores de janelas e/ou displays integrados podem ser implementados em muitos tipos de janelas (por exemplo, a janela (s) em um veículo, edifício, avião, barco, bonde, trem, etc.).

[00245] A figura 19 é um diagrama que ilustra um exemplo de zonas 1985a-b em uma janela 1979. Neste exemplo, a zona A 1985a é uma seção lateral (por exemplo, a seção do lado esquerdo) da janela 1979 (por exemplo, para-brisas) e a zona B 1985b é outra seção lateral (por exemplo, a seção lateral direita) da janela. Supondo que o indicador 1979 é um para-brisa de um veículo, por exemplo, o dispositivo eletrônico 1651 (por exemplo, renderizador multizona 1663) pode renderizar um maior nível de detalhe na zona correspondente ao lado do condutor (por exemplo, a zona A, 1985a). Um nível mais baixo de detalhe pode ser renderizado na zona correspondente ao lado do passageiro (por exemplo, a zona B 1985b). Deste modo, o condutor pode visualizar maior nível de detalhe de uma imagem de uma área que o controlador tipicamente vê. Deve ser notado que um número diferente de zonas pode ser processado. Adicionalmente ou em alternativa,

as zonas 1985a-b podem, cada uma, ocupar metade ou diferentes proporções da janela 1979 (por exemplo, 70% versus 30%).

[00246] A figura 20 é um diagrama que ilustra um outro exemplo de zonas 2085a-c em um indicador 2079. Neste exemplo, a zona A 2085a é uma barra horizontal através de uma parte central da janela 2079 (por exemplo, para-brisas). Zona B 2085b inclui duas barras horizontais que fazem fronteira com a zona A 2085a e a zona C 2085c inclui duas barras horizontais que fazem fronteira com zona B 2085b. Supondo que o indicador 2079 é um para-brisa de um veículo, por exemplo, o dispositivo eletrônico 1651 (por exemplo, renderizador multizona 1663) pode renderizar um maior nível de detalhe na zona A 2085a. Um nível inferior de detalhe pode ser renderizado na zona B 2085b. Um nível ainda menor de detalhe pode ser renderizado na zona C 2085c. Desta forma, o condutor pode visualizar maior nível de detalhe de uma imagem em uma área que o condutor visualiza ao olhar sobre a extensão do para-brisa. Deve ser notado que um número diferente de zonas pode ser processado. Adicionalmente ou em alternativa, as zonas 2085a-c podem cada ocupar proporções iguais ou diferentes da janela 2079 (por exemplo, 50% / 30% / 20%).

[00247] A figura 21 é um diagrama que ilustra um outro exemplo de zonas 2185a-c em múltiplas janelas 2179a-e. Neste exemplo, a zona A 2185a é renderizada na janela A 2179a (por exemplo, um para-brisa). Zona B 2185b é renderizada na 2179b janela B e janela C 2179c (por exemplo, janelas frontais esquerda e direita). Zona C 2185c é renderizada na janela D 2179d e janela E 2179e (por exemplo, janelas traseiras esquerda e direita). Assumindo que janela

A 2179a é um para-brisa de um veículo, por exemplo, o dispositivo eletrônico 1651 (por exemplo, renderizador multizona 1663) pode renderizar um maior nível de detalhe na zona A 2185a. Um nível inferior de detalhe pode ser renderizado na zona B 2185b. Um nível ainda menor de detalhe pode ser renderizado na zona C 2185c. Desta forma, o condutor pode visualizar maior nível de detalhe de uma imagem em uma área que o condutor visualiza quando se olha para fora do para-brisa. Deve ser notado que uma configuração diferente de zonas e/ou um número diferente de zonas pode ser processado. Por exemplo, a zona C 2185c (ou outra zona) pode ser processada em uma janela traseira.

[00248] A figura 22 é um diagrama de fluxo que ilustra uma configuração mais específica de um método 2200 para renderizar múltiplas zonas. O método 2200 pode ser executado pelo dispositivo eletrônico 1651 descrito em conexão com a Figura 16.

[00249] O dispositivo eletrônico 1651 pode executar 2202 rastreamento ocular. Isto pode ser conseguido como descrito em ligação com uma ou mais das Figuras 3, 5 e 16. Por exemplo, o dispositivo eletrônico 1651 pode captar uma ou mais imagens de um usuário (por exemplo, um ou mais usuários). O dispositivo eletrônico 1651 pode identificar um ou mais olhos de um ou mais usuários na imagem (s). O dispositivo eletrônico 1651 pode então determinar para onde o olho (s) de um ou mais usuários está direcionado em relação a um ou mais displays (por exemplo, display 1673) e/ou imagem (s) no display (s).

[00250] O dispositivo eletrônico 1651 pode renderizar 2204 uma primeira zona de uma imagem baseado em

um rastreamento ocular. Isto pode ser conseguido como descrito em ligação com uma ou mais das Figuras 1-5 e 8-17. Por exemplo, o dispositivo eletrônico 1651 pode executar uma ou mais de busca de geometria (por exemplo, busca de vértice), shader de vértice, shader de controle de tecelagem, tecelagem, shader de avaliação de tecelagem, shader de geometria, mosaico, rasterização e/ou shader de fragmento correspondente a uma primeira zona de uma imagem. A localização da primeira zona pode ser baseada no rastreamento ocular. Por exemplo, o dispositivo eletrônico 1651 pode estabelecer a primeira zona como uma seção de uma imagem em torno do local para onde o olho (s) do usuário está direcionado. Em algumas configurações, a primeira zona pode ser uma área circular (ou uma área de outro formato) aproximadamente centrada no ponto de visualização (por exemplo, imagem) para onde o usuário esta olhando. Em outras configurações, a primeira zona pode ser toda a área de uma imagem que corresponde a uma janela que o usuário está visualizando.

[00251] O dispositivo eletrônico 1651 pode renderizar 2206 uma segunda zona da imagem com base no rastreamento ocular. Isto pode ser conseguido como descrito em ligação com uma ou mais das Figuras 1-5 e 8-17. Por exemplo, o dispositivo eletrônico 1651 pode executar um ou mais de busca de geometria (por exemplo, busca de vértice), shader de vértice, shader de controle de tecelagem, tecelagem, shader de avaliação de tecelagem, shader de geometria, mosaicos, rasterização e/ou shader de fragmento correspondente a uma segunda zona da imagem. A primeira zona pode ter um nível tecelado de detalhe mais elevado do que a

segunda zona. Por exemplo, a primeira zona pode ter uma configuração de tecelagem mais elevada (por exemplo, uma densidade de geometria mais elevada) do que uma configuração de tecelagem para a segunda zona. A localização da segunda zona pode ser baseada em um rastreamento ocular. Em algumas configurações, o dispositivo eletrônico 1651 para estabelecer a segunda zona como uma seção de uma imagem em torno da primeira zona. Por exemplo, a segunda zona pode ser uma área circular (ou uma região de outro formato), que envolve a primeira zona. Em outras configurações, a segunda zona pode ser uma área de uma imagem que corresponde a uma janela que o usuário não está visualizando. Em ainda outras configurações, a segunda zona pode ser a imagem inteira, exceto para a primeira zona. Note-se que mais zonas podem ser renderizadas. As zonas podem ou não sobrepor-se. Em algumas configurações, a renderização de zona pode ser priorizada como descrito em conexão com a Figura 16.

[00252] O dispositivo eletrônico 1651 pode apresentar 2208 a primeira zona e a segunda zona da janela em pelo menos um veículo. Isto pode ser conseguido tal como descrito acima em ligação com uma ou mais das Figuras 16-21. Por exemplo, a primeira zona e a segunda zona podem ser apresentadas em seções separadas de uma janela (por exemplo, para-brisas) e/ou em diferentes janelas.

[00253] A figura 23 é um diagrama que ilustra um outro exemplo de zonas 2385a-c em um indicador 2379. Neste exemplo, a zona A 2385a é uma área circular da janela 2379 (por exemplo, para-brisas), onde um usuário está olhando com base no rastreamento ocular. Zona B 2385b é uma área circular em torno da zona A 2385a. Zona C 2385c é uma área da janela

2379 que não inclui zona A 2385a e zona B 2385b. Supondo que a janela 2379 é um para-brisa de um veículo, por exemplo, o dispositivo eletrônico 1651 (por exemplo, renderizador multizona 1663) pode renderizar um maior nível de detalhe na zona A 2385a. Um nível inferior de detalhe pode ser renderizado na zona B 2385b. Um nível ainda menor de detalhe pode ser renderizado na zona C 2385c. Desta forma, o condutor pode ver de maior nível de detalhe de uma imagem em uma área onde o condutor está atualmente olhando (em qualquer lugar no para-brisa). Deve ser notado que um número diferente de zoas pode ser renderizado.

[00254] A Figura 24 ilustra certos componentes que podem ser incluídos dentro de um dispositivo eletrônico 2487. O dispositivo eletrônico 2487 pode ser (ou pode ser incluído dentro de) uma câmara, câmara de vídeo, câmara digital, telefone celular, smartphone, computador (por exemplo, computador desktop, computador laptop, etc.), tablet, reproduutor de mídia, televisão, veículo, fone de ouvido (por exemplo, fone de ouvido de realidade virtual, fone de ouvido de realidade aumentada), display montado na cabeça, câmera pessoal, câmera de ação, câmera de vigilância, câmera montada, câmera conectada, robô, aviões, drone, veículo aéreo não tripulado (UAV), equipamentos de saúde, console de jogos, assistentes digitais pessoais (PDA), set-top box, etc. O dispositivo eletrônico 2487 pode ser implementado de acordo com um ou mais dos dispositivos eletrônicos 102, 302, 866, 902, 1066, 1102, 1266, 1302, 1466, 1502, 1651 aqui descrito.

[00255] O dispositivo eletrônico 2487 inclui um processador 2407. O processador 2407 pode ser um

microprocessador de finalidade geral simples ou multichip (por exemplo, um ARM), um microprocessador de finalidade especial (por exemplo, um processador de sinal digital (DSP)), um microcontrolador, um arranjo de portas programável, etc. O processador 2407 pode ser referido como uma unidade de processamento central (CPU) e/ou circuito VLIW (por exemplo, GPU). O processador 2407 pode ser implementado de acordo com um ou mais dos circuitos VLIW 104, 304, 868, 904, 1068, 1104, 1268, 1304, 1468, 1504 e/ou processadores 324, 1015, 1124, 1661 aqui descritos. Embora apenas um único processador 2407 seja mostrado no dispositivo eletrônico 2487, em uma configuração alternativa, poderia ser usada uma combinação de processadores (por exemplo, um ARM e DSP, uma CPU e GPU, etc.).

[00256] O dispositivo eletrônico 2487 também inclui memória 2489. A memória 2489 pode ser qualquer componente eletrônico capaz de armazenar informação eletrônica. A memória 2489 pode ser incorporada como memória de acesso aleatório (RAM), memória somente de leitura (ROM), meios de armazenamento em disco magnético, meios de armazenamento ópticos, dispositivos de memória flash em RAM, memória on-board incluída com o processador, memória EPROM, memória EEPROM, registradores, e assim por diante, incluindo suas combinações. A memória 2489 pode ser implementada de acordo com uma ou mais das memórias 110, 114, 310, 314, 803, 805, 910, 914, 1003, 1005, 1110, 1114, 1203, 1205, 1310, 1314, 1403, 1405, 1510, 1514, 1667 aqui descritas.

[00257] Dados 2493a e instruções 2491a podem ser armazenados na memória 2489. As instruções 2491a podem ser executáveis pelo processador 2407 para implementar um ou

mais dos métodos 200, 500, 600, 1700, 2200 aqui descritos. Executar as instruções 2491a pode envolver o uso dos dados 2493a que são armazenados na memória 2489. Quando o processador 2407 executa as instruções 2491, várias porções das instruções 2491b podem ser carregadas para o processador 2407, e várias porções de dados podem 2493b ser carregadas para o processador 2407.

[00258] O dispositivo eletrônico 2487 também pode incluir um transmissor 2497 e um receptor 2499 para permitir a transmissão e recepção de sinais de e para o dispositivo eletrônico 2487. O transmissor 2497 e o receptor 2499 podem ser coletivamente referidos como um transceptor 2401. Uma ou múltiplas antenas 2409a-b podem ser eletricamente acopladas ao transceptor 2401. O dispositivo eletrônico 2487 pode também incluir (não mostrado) múltiplos transmissores, múltiplos receptores, múltiplos transceptores e/ou antenas adicionais.

[00259] O dispositivo eletrônico 2487 pode incluir um processador de sinal digital (DSP) 2403. O dispositivo eletrônico 2487 também pode incluir uma interface de comunicação 2405. A interface de comunicação 2405 pode permitir um ou mais tipos de entrada e/ou saída. Por exemplo, a interface de comunicação 2405 pode incluir uma ou mais portas e/ou dispositivos de comunicação para ligar outros dispositivos ao dispositivo eletrônico 2487. Adicionalmente ou em alternativa, a interface de comunicação 2405 pode incluir uma ou mais interfaces (por exemplo, tela de toque, teclado, teclado, microfone, câmara, etc.). Por exemplo, a interface de comunicação 2405 pode permitir que um usuário interaja com o dispositivo eletrônico 2487.

[00260] Os diversos componentes do dispositivo eletrônico 2487 podem ser acoplados um ao outro por um ou mais barramentos, os quais podem incluir um barramento de alimentação, uma barramento de sinais de controle, um barramento de sinal de status, um barramento de dados, etc. Por uma questão de clareza, os vários barramento são ilustrados na Figura 24 como um sistema de barramento 2495.

[00261] O termo "determinação" engloba uma larga variedade de ações e, portanto, "determinar" pode incluir calcular, computar, renderizar, derivar, investigar, buscar (por exemplo, buscar em uma tabela, um banco de dados ou outra estrutura de dados), averiguar e similares. Além disso, "determinar" pode incluir receber (por exemplo, receber informações), acessar (por exemplo, acessar dados em uma memória) e similares. Além disso, "determinar" pode incluir resolver, selecionar, escolher, estabelecer e similares.

[00262] A frase "com base em" não significa "baseado apenas em" a menos que expressamente indicado em contrário. Em outras palavras, a frase "com base em" descreve tanto "com base apenas em" quanto "com base pelo menos em".

[00263] O termo "processador" deve ser interpretado de forma ampla para abranger um processador de uso geral, uma unidade de processamento central (CPU), um microprocessador, um processador de sinal digital (DSP), uma unidade de processamento gráfico (GPU), um controlador, um microcontrolador, uma máquina de estado, e assim por diante. Sob algumas circunstâncias, um "processador" pode referir-se a um circuito integrado de aplicação específica (ASIC), um dispositivo lógico programável (PLD), um arranjo de porta programável em campo (FPGA), etc. O termo "processador" pode

referir-se a uma combinação de dispositivos de renderização, por exemplo, uma combinação de um DSP e um microprocessador, uma pluralidade de microprocessadores, um ou mais microprocessadores em conjunto com um núcleo de DSP, uma combinação de uma CPU e GPU, ou qualquer outro tipo de configuração.

[00264] O termo "memória" deve ser interpretado de forma ampla para abranger qualquer componente eletrônico capaz de armazenar informação eletrônica. O termo memória pode se referir a vários tipos de mídia legível por processador, tais como memória de acesso aleatório (RAM), memória somente de leitura (ROM), memória de acesso aleatório não volátil (NVRAM), memória somente leitura programável (PROM), memória somente de leitura programável apagável (EPROM), PROM eletricamente apagável (EEPROM), memória flash, armazenamento de dados magnéticos ou ópticos, registradores, etc. Memória é referida estar em comunicação eletrônica com um processador, se o processador puder ler informação a partir de e/ou gravar informações na memória. Memória que é parte integrante de um processador está em comunicação eletrônica com o processador.

[00265] Os termos "instruções" e "código" devem ser interpretados de forma ampla para incluir qualquer tipo de declaração (s) legível por computador. Por exemplo, os termos "instruções" e "código" podem se referir a um ou mais programas, rotinas, sub-rotinas, funções, procedimentos, etc. "Instruções" e "código" podem incluir uma declaração legível por computador única ou muitas declarações legíveis por computador.

[00266] As funções aqui descritas podem ser implementadas em software ou firmware sendo executados por hardware. As funções podem ser armazenadas como uma ou mais instruções em um meio legível por computador. Os termos "meio legível por computador" ou "produto de programa de computador" referem-se a qualquer meio de armazenamento tangível que pode ser acessado por um computador ou um processador. A título de exemplo, e não como limitação, um meio legível por computador podem compreender RAM, ROM, EEPROM, CD-ROM ou outro armazenamento em disco óptico, armazenamento em disco magnético ou outros dispositivos de armazenamento magnéticos, ou qualquer outro meio que possa ser utilizado para portar ou armazenar código de programa desejado na forma de instruções ou estruturas de dados e que pode ser acessado por um computador. Disco e disquete, como aqui utilizado, incluem disco compacto (CD), disco laser, disco óptico, disco versátil digital (DVD), disco flexível e disco Blu-ray® onde disquetes geralmente reproduzem dados magneticamente, enquanto que discos reproduzem dados opticamente com lasers. Note-se que um meio legível por computador pode ser tangível e não transitório. O termo "produto de programa de computador" refere-se a um dispositivo de computação ou processador, em combinação com o código ou instruções (por exemplo, um "programa") que podem ser executados, processados ou calculados pelo dispositivo de computação ou processador. Tal como aqui utilizado, o termo "código" pode referir-se a software, instruções, ou código de dados que são executáveis por um dispositivo de computação ou processador.

[00267] Software ou instruções podem também ser transmitidos através de um meio de transmissão. Por exemplo, se o software é transmitido de um site, servidor ou outra fonte remota utilizando um cabo coaxial, cabo de fibra óptica, par trançado, linha de assinante digital (DSL) ou tecnologias sem fio tais como infravermelhos, rádio e micro-ondas, então cabo coaxial, cabo de fibra óptica, par trançado, DSL, ou tecnologias sem fio, tais como infravermelho, rádio e micro-ondas estão incluídos na definição de meio de transmissão.

[00268] Os métodos aqui divulgados compreendem uma ou mais etapas ou ações para alcançar o método descrito. As etapas e/ou ações de método podem ser trocadas umas com as outras sem se afastar do âmbito das reivindicações. Em outras palavras, a não ser que seja necessário um fim específico de etapas ou ações para uma operação adequada do método que está sendo descrito, a ordem e/ou o uso de etapas e/ou ações específicas podem ser modificados sem se afastar do âmbito das reivindicações.

[00269] Além disso, deve ser notado que os módulos e/ou outros meios adequados para a realização dos métodos e técnicas aqui descritos, podem ser baixados e/ou obtidos de outro modo por um dispositivo. Por exemplo, um dispositivo pode ser acoplado a um servidor para facilitar a transferência de meios para a realização dos métodos aqui descritos. Alternativamente, vários métodos aqui descritos podem ser providos através de um meio de armazenamento (por exemplo, memória de acesso aleatório (RAM), memória somente de leitura (ROM), um meio de armazenamento físico tal como um disco compacto (CD) ou disquete, etc.), de tal modo que

um dispositivo pode obter os vários métodos ao se acoplar ou prover meios de armazenamento para o dispositivo.

[00270] Deve ser compreendido que as reivindicações não se limitam à configuração precisa e componentes ilustrados acima. Várias modificações, alterações e variações podem ser feitas no arranjo, operação e detalhes dos sistemas, métodos e aparelhos aqui descritos, sem se afastar do escopo das reivindicações.

### REIVINDICAÇÕES

1. Aparelho **caracterizado** pelo fato de que comprehende:

meios para renderizar uma primeira zona de uma imagem com um primeiro nível de detalhe durante processamento de tecelação,

meios para renderizar uma segunda zona da imagem com um segundo nível de detalhe durante processamento de tecelação, em que o primeiro nível de detalhe é maior que o segundo nível de detalhe; e

meios para apresentar a primeira zona e a segunda zona em pelo menos uma tela (display),

em que os meios para renderizar a primeira e segunda zonas são configurados para renderizar a primeira e segunda zonas em uma única chamada de desenho.

2. Aparelho, de acordo com a reivindicação 1, **caracterizado** pelo fato de que os meios para apresentar comprehendem meios para enviar a primeira zona e a segunda zona para pelo menos um display para apresentar a primeira zona e a segunda zona; ou

compreendendo adicionalmente meios para apresentar a primeira zona em uma primeira janela de veículo e para apresentar a segunda zona em uma segunda janela de veículo.

3. Aparelho, de acordo com a reivindicação 1, **caracterizado** pelo fato de que o pelo menos um display é um projetor de janela ou um painel de visualização de janela integrado.

4. Aparelho, de acordo com a reivindicação 1, **caracterizado** pelo fato de que os meios para apresentar são

configurados para apresentar a primeira zona em uma primeira janela de veículo e para apresentar a segunda zona em uma segunda janela veículo.

5. Aparelho, de acordo com a reivindicação 4, caracterizado pelo fato de que a primeira janela é um para-brisa dianteiro e a segunda janela é outra janela de um veículo.

6. Aparelho, de acordo com a reivindicação 2, caracterizado pelo fato de que os meios para apresentar são configurados para apresentar a primeira zona em uma primeira seção de uma janela de veículo e para apresentar a segunda zona em uma segunda seção da janela de veículo.

7. Aparelho, de acordo com a reivindicação 6, caracterizado pelo fato de que:

a primeira seção é uma seção lateral de um para-brisa e a segunda seção é outra seção lateral do para-brisa; ou

em que a primeira seção é uma primeira barra horizontal através de um para-brisa e a segunda seção é uma segunda barra horizontal através do para-brisa.

8. Aparelho, de acordo com a reivindicação 1, caracterizado pelo fato de que compreende adicionalmente:

meios para realizar rastreamento ocular para múltiplos usuários incluindo um primeiro usuário e um segundo usuário; e

meios para priorizar a renderização da primeira e segunda zonas, com base no rastreamento ocular, em que a priorização da renderização da primeira e segunda zonas compreende renderizar a primeira zona com um primeiro nível de detalhe que é maior que o segundo nível de detalhe para

a segunda zona com base na primeira zona sendo para apresentação para o primeiro usuário e a segunda zona sendo para apresentação para o segundo usuário e o primeiro usuário sendo priorizado em relação ao segundo usuário.

9. Aparelho, de acordo com a reivindicação 1, **caracterizado** pelo fato de que o aparelho é pelo menos parte de um módulo removível em um veículo.

10. Aparelho, de acordo com a reivindicação 1, **caracterizado** pelo fato de que o aparelho é pelo menos parte de um dispositivo de realidade virtual, VR, como um headset VR.

11. Aparelho, de acordo com a reivindicação 1, **caracterizado** pelo fato de que os meios para renderizar a primeira e segunda zonas são configurados para acessar de forma assíncrona memória para renderizar os níveis diferentes de detalhes na única chamada de desenho.

12. Aparelho, de acordo com a reivindicação 1, **caracterizado** pelo fato de que os meios para renderizar a primeira e segunda zonas compreendem um controlador de buffer de quadro não uniforme configurado para acessar diferentes subconjuntos de um ou mais buffers de quadro, os diferentes conjuntos do buffer de quadro correspondendo a diferentes níveis de detalhes.

13. Aparelho, de acordo com a reivindicação 1, **caracterizado** pelo fato de que os meios para renderizar a primeira e segunda zonas compreendem um circuito de palavra de instrução muito longa, VLIW.

14. Método realizado por um dispositivo eletrônico, o método **caracterizado** pelo fato de que compreende:

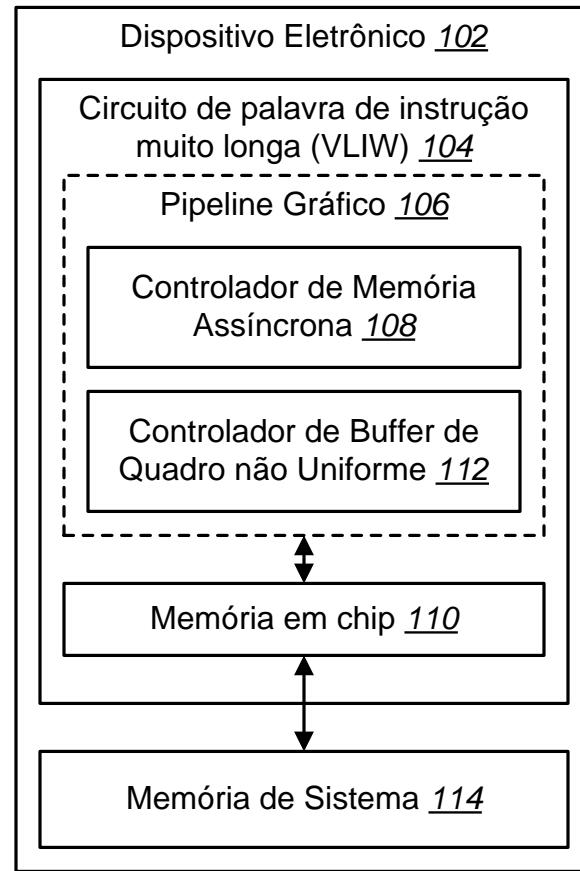
renderizar uma primeira zona de uma imagem com um primeiro nível de detalhe durante processamento de tecelação;

renderizar uma segunda zona da imagem com um segundo nível de detalhe durante processamento de tecelação, em que o primeiro nível de detalhe é maior que o segundo nível de detalhe; e

apresentar a primeira zona e a segunda zona em pelo menos uma tela (display),

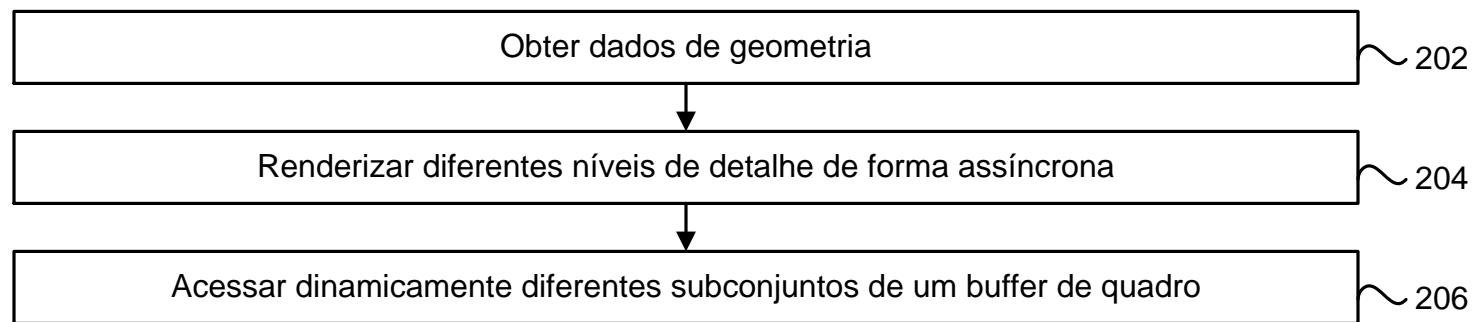
em que o dispositivo eletrônico renderiza a primeira e segunda zonas durante uma única chamada de desenho.

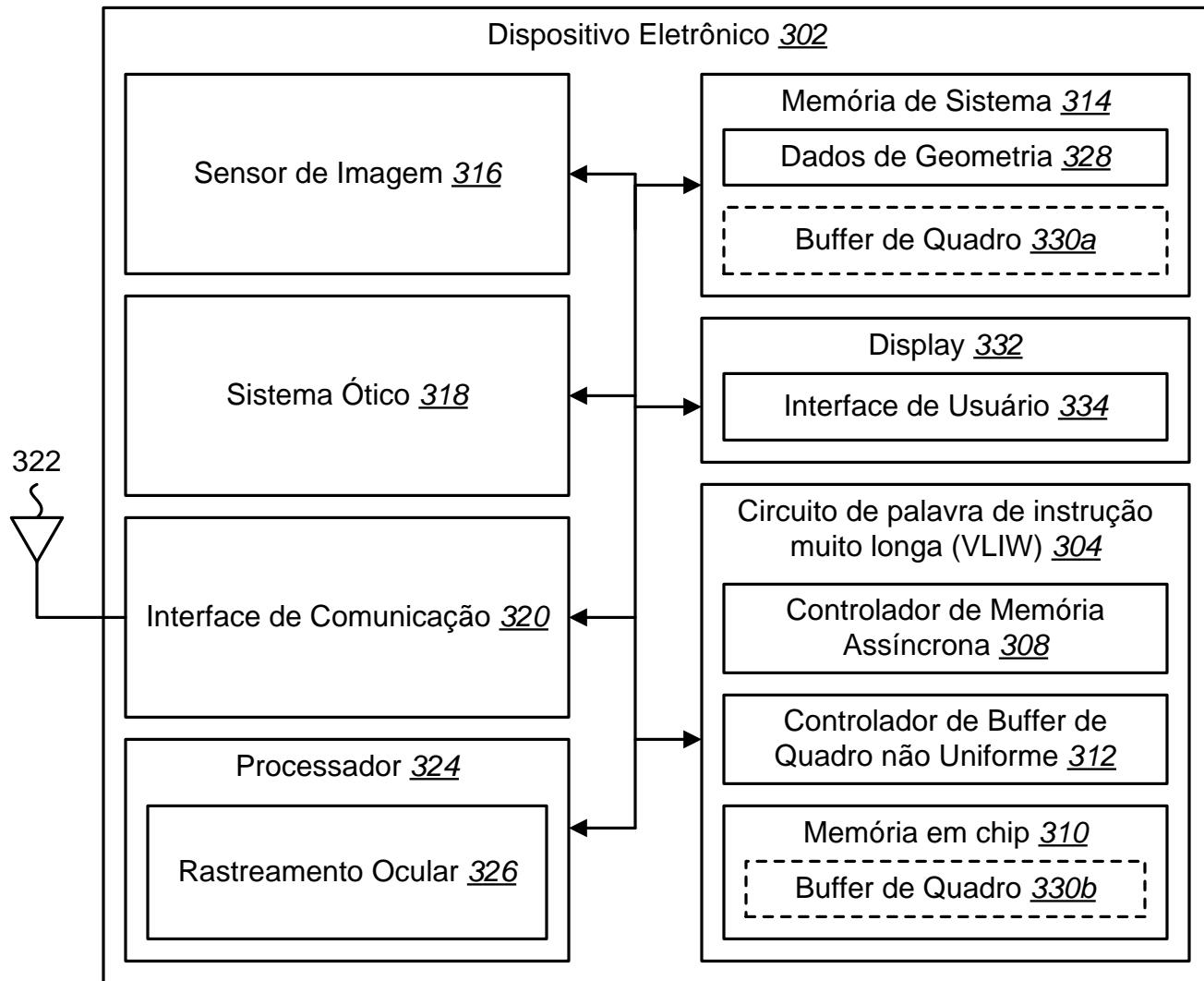
15. Memória legível por computador **caracterizada** pelo fato de que possui instruções nela armazenadas que, quando executadas, fazem com que um computador realize o método conforme definido na reivindicação 14.

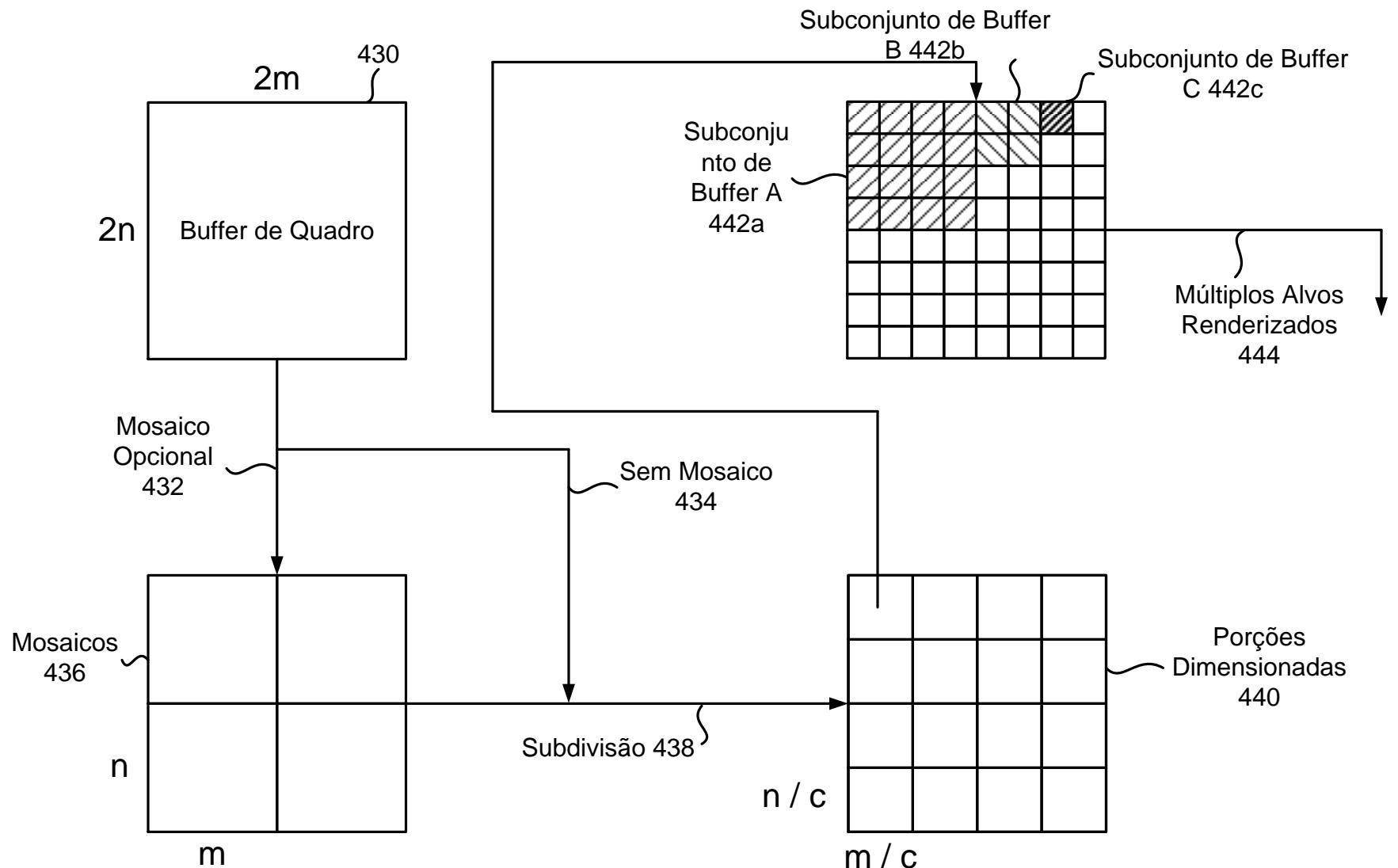


**FIG. 1**

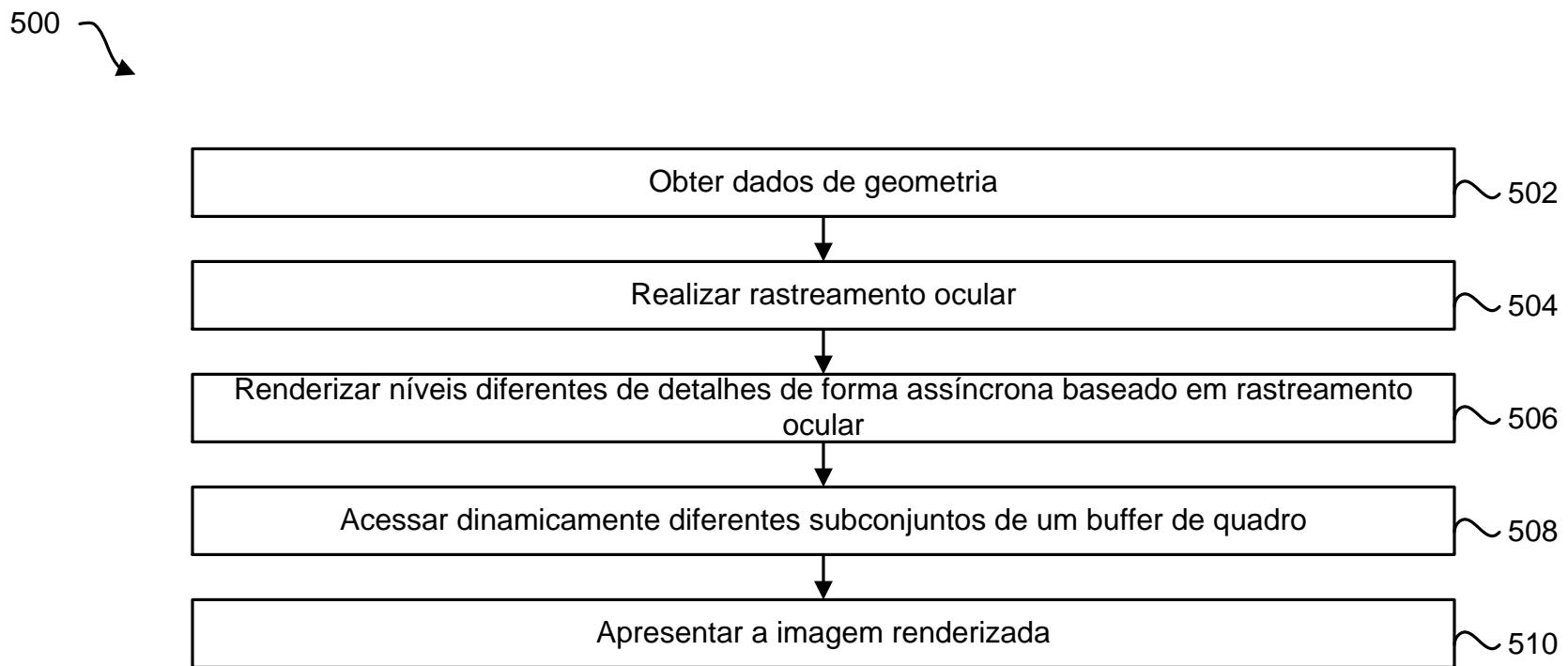
200

**FIG. 2**

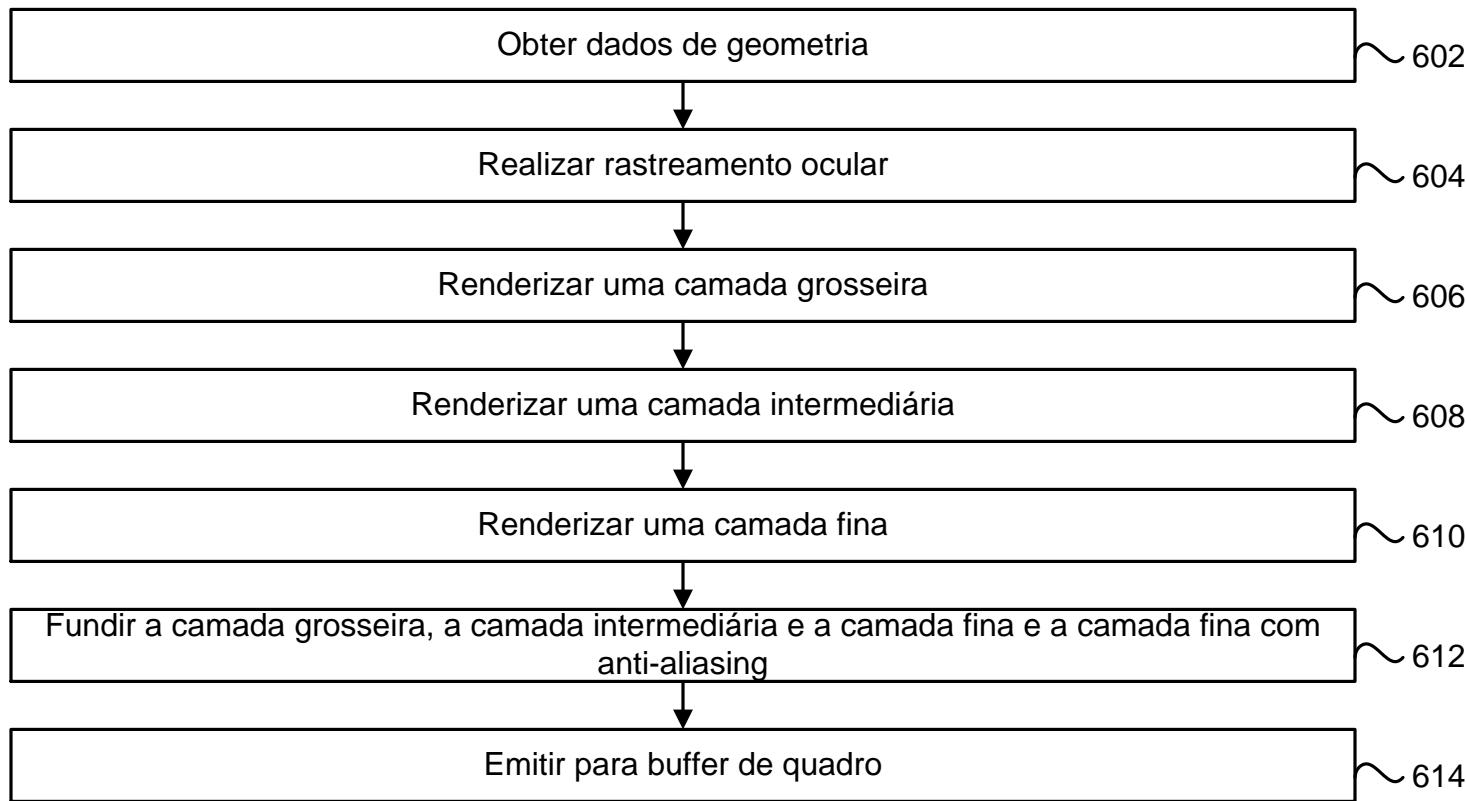
**FIG. 3**



**FIG. 4**

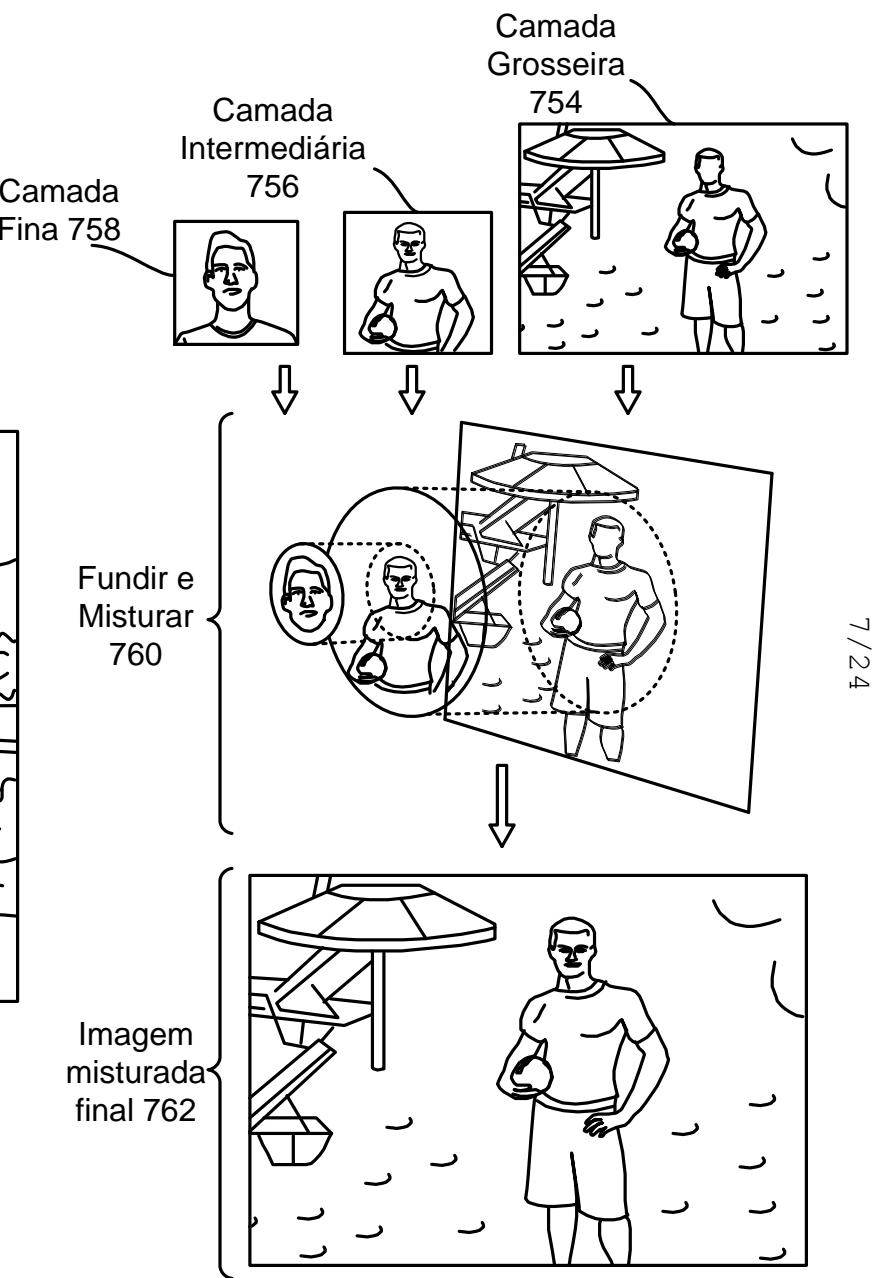
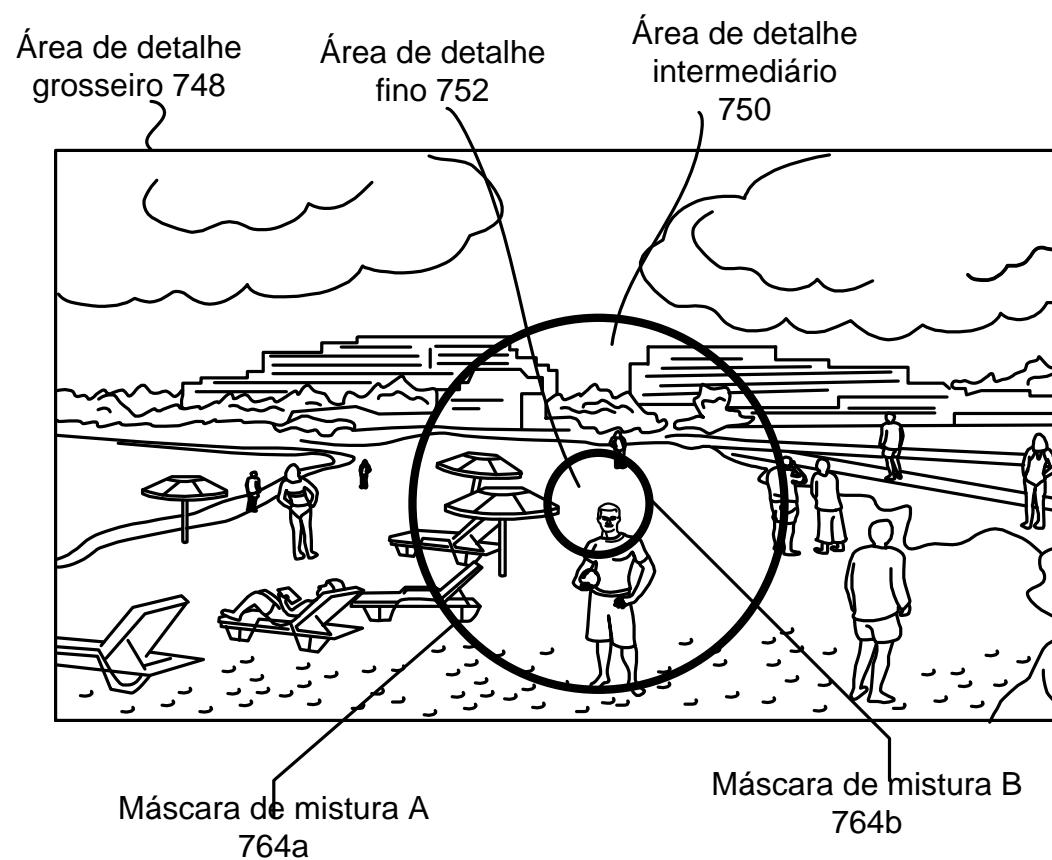
**FIG. 5**

600  
~~~~~



6 / 24

**FIG. 6**

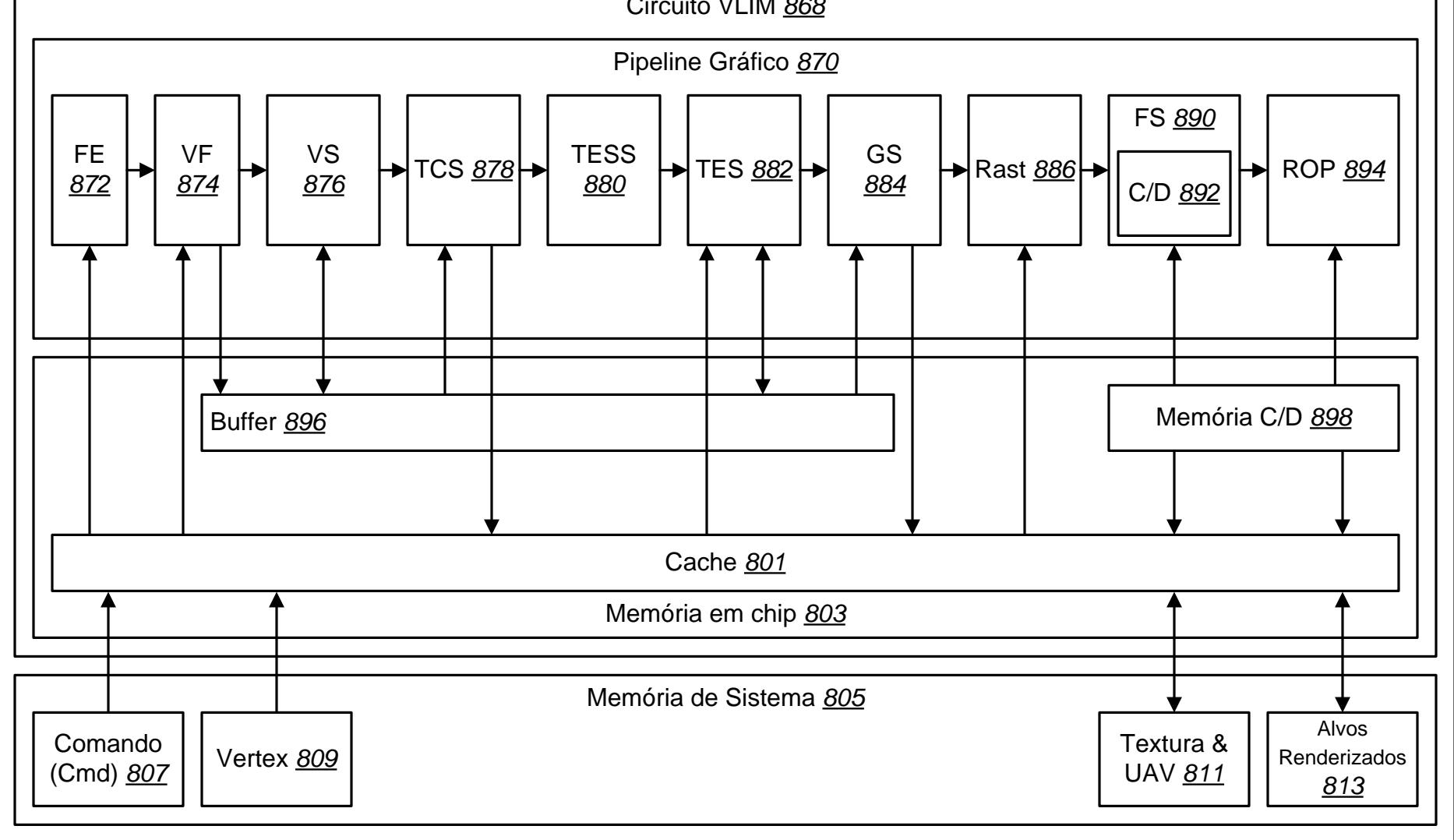


**FIG. 7**

Dispositivo Eletrônico 866

Círcuito VLIM 868

Pipeline Gráfico 870



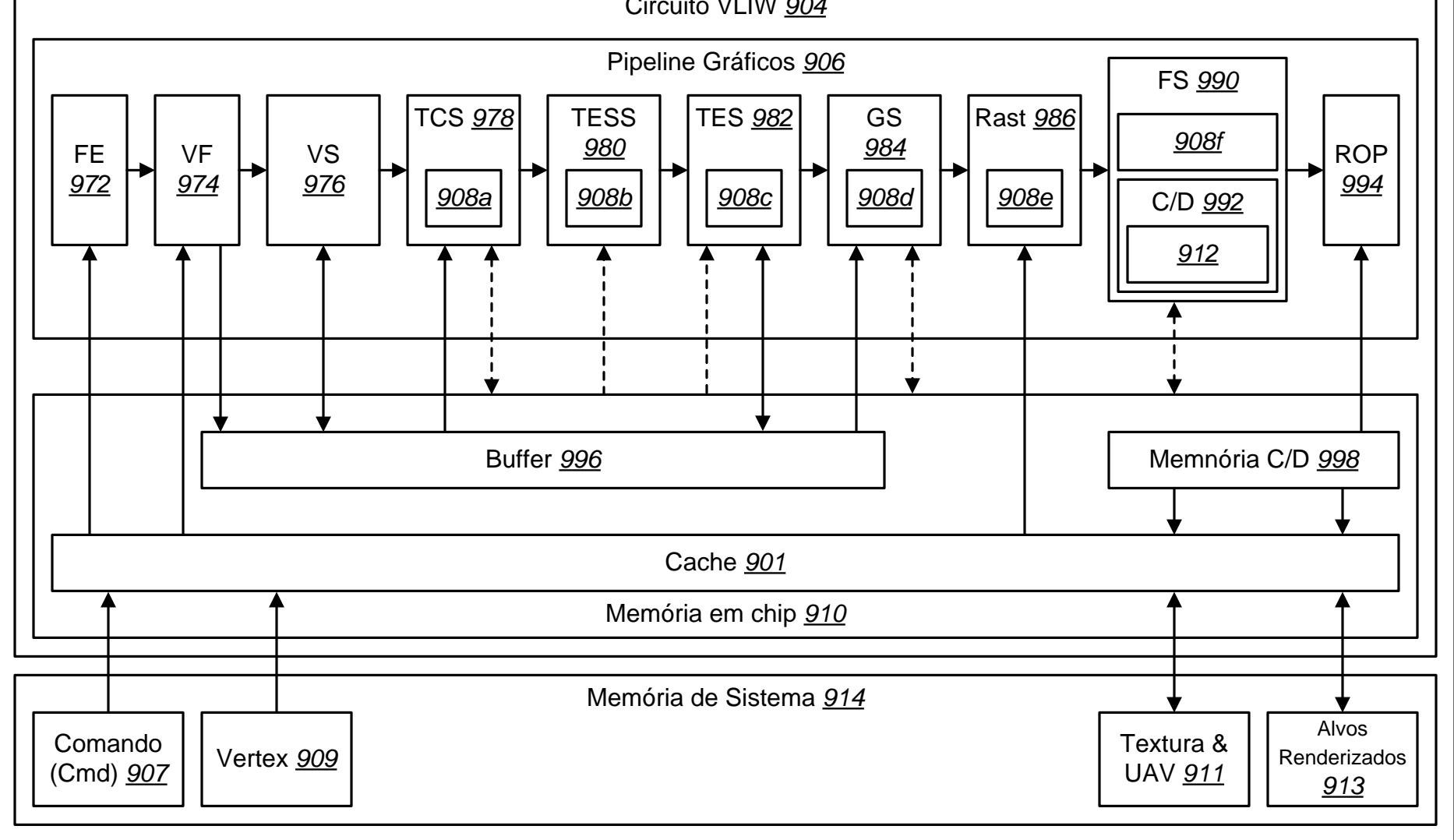
8 / 24

**FIG. 8**

Dispositivo Eletrônico 902

Círculo VLIW 904

Pipeline Gráficos 906



9 / 24

**FIG. 9**

Dispositivo Eletrônico 1066

Memória de Sistema 1005

Processador 1015

Círcuito VLIW 1068

Pipeline Gráficos 1070

CS  
1017

VF  
1074

VS  
1076

HS  
1078

TESS  
1080

DS  
1082

GS  
1084

SOL  
1019

Clip  
1021

Rast  
1086

FS  
1090

C/D  
1092

Cache 1001

Memória em chip 1003

10 / 24

**FIG. 10**

Dispositivo Eletrônico 1102

Memória de Sistema 1114

Processador 1124

Círcuito VLIW 1104

Pipeline Gráficos 1106

CS 1117

VF 1174

VS 1176

HS 1178  
1108a

TESS 1180  
1108b

DS 1182  
1108c

GS 1184  
1108d

SOL 1119

Clip 1121

Rast 1186  
1108e

FS 1190  
1108f

C/D 1192  
1112

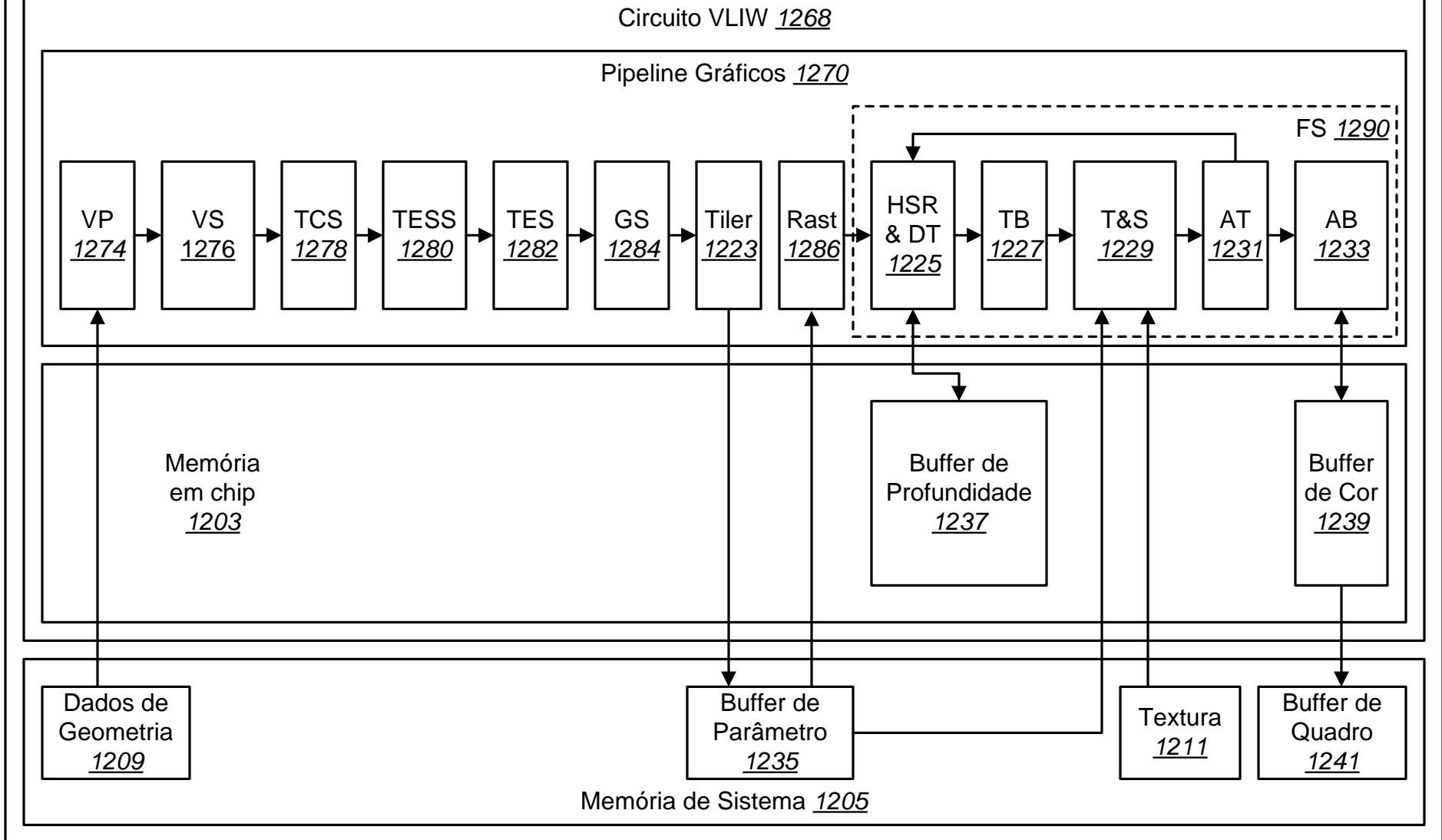
Cache 1101

**FIG. 11**

Dispositivo Eletrônico 1266

Círcuito VLIW 1268

Pipeline Gráficos 1270



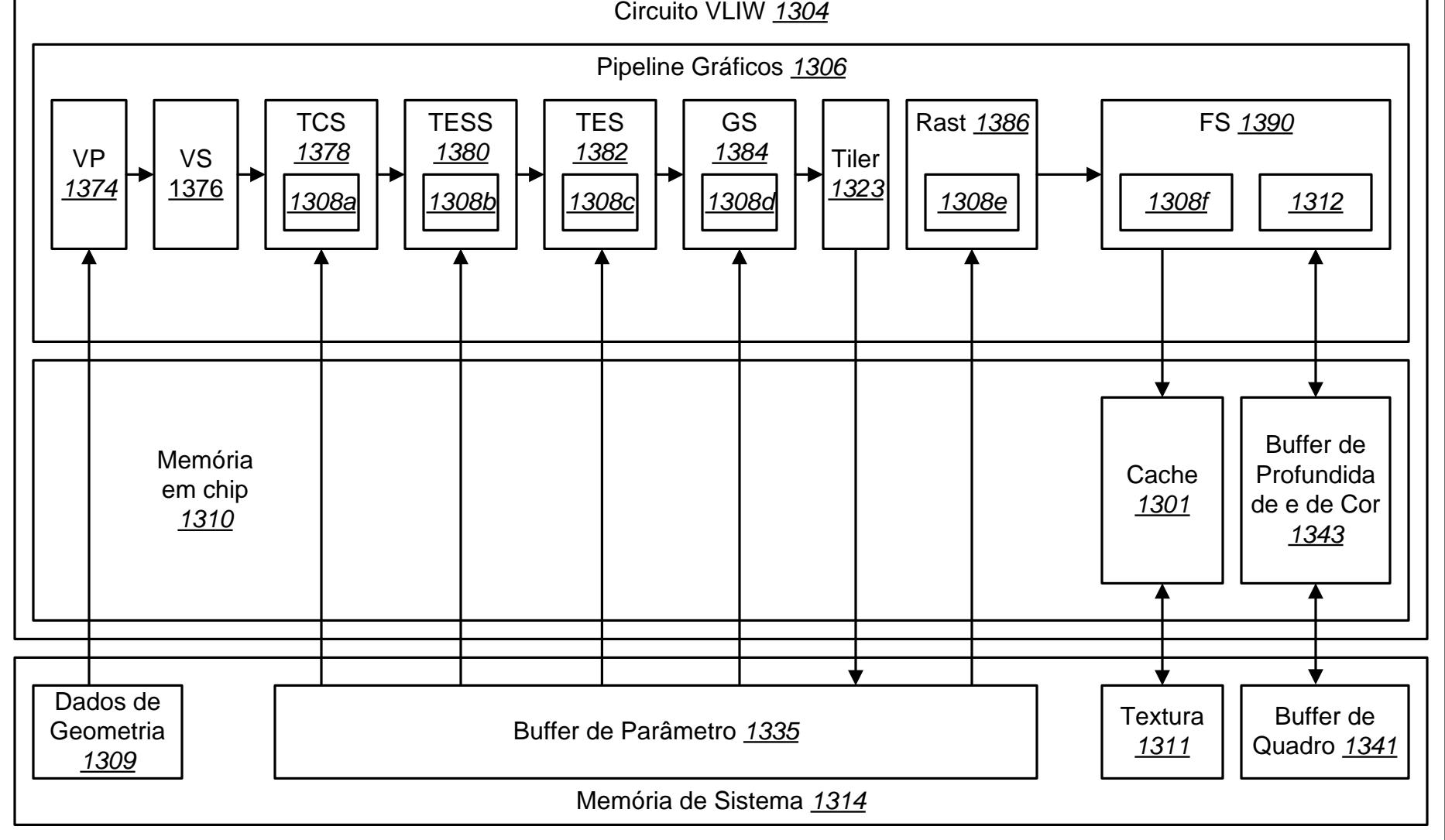
12 / 24

**FIG. 12**

Dispositivo Eletrônico 1302

Círcuito VLIW 1304

Pipeline Gráficos 1306



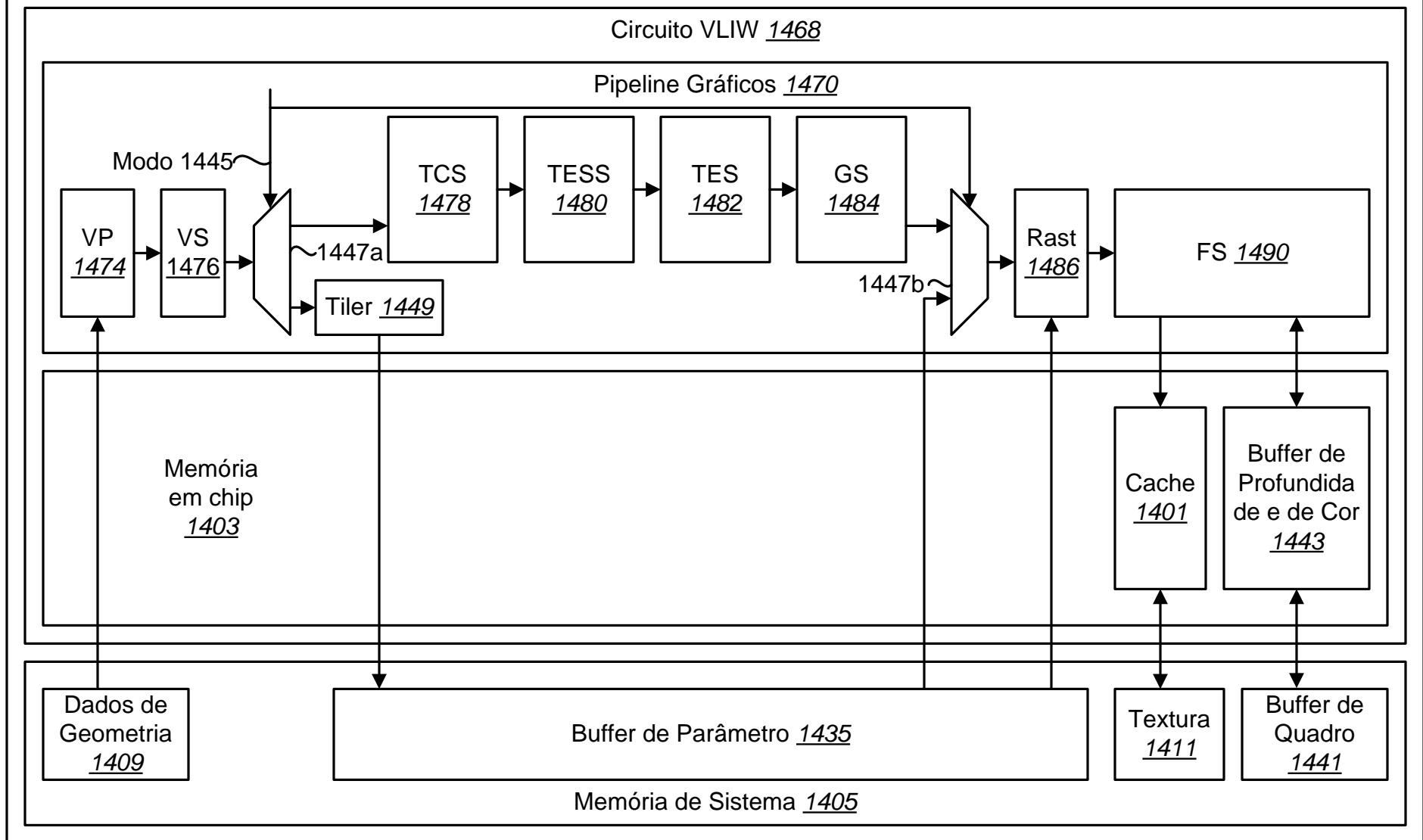
13 / 24

**FIG. 13**

Dispositivo Eletrônico 1466

## Circuito VLIW 1468

## Pipeline Gráficos 1470

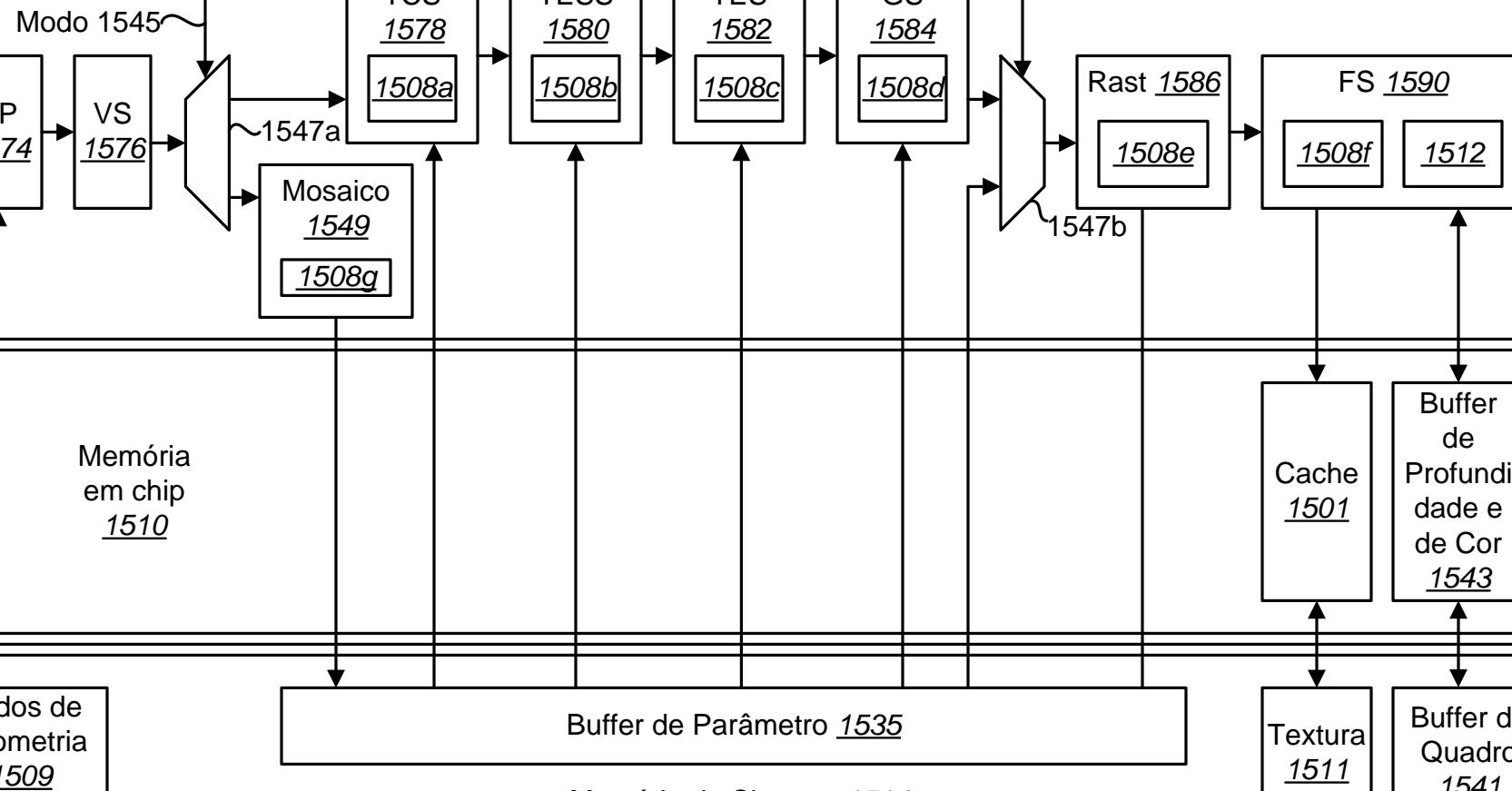


## FIG. 14

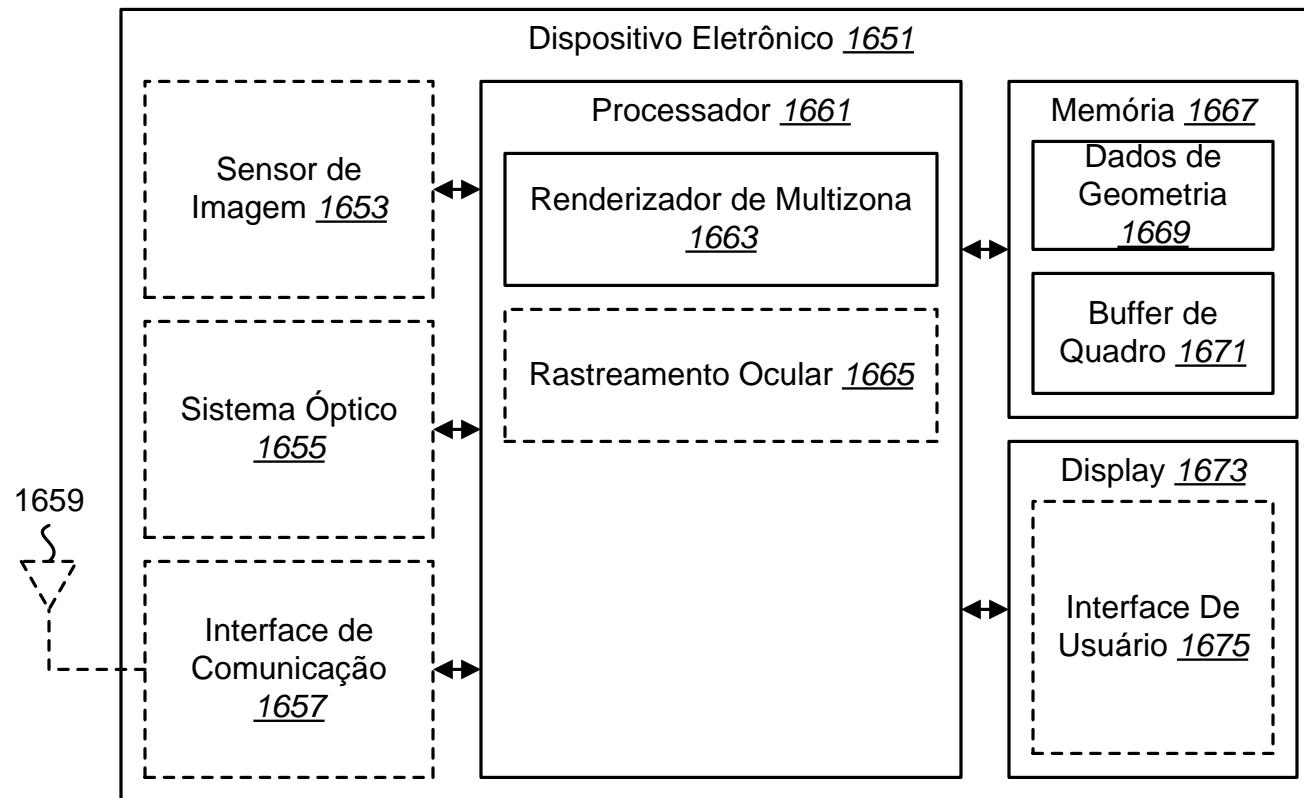
Dispositivo Eletrônico 1502

Círcuito VLIW 1504

Pipeline Gráficos 1506

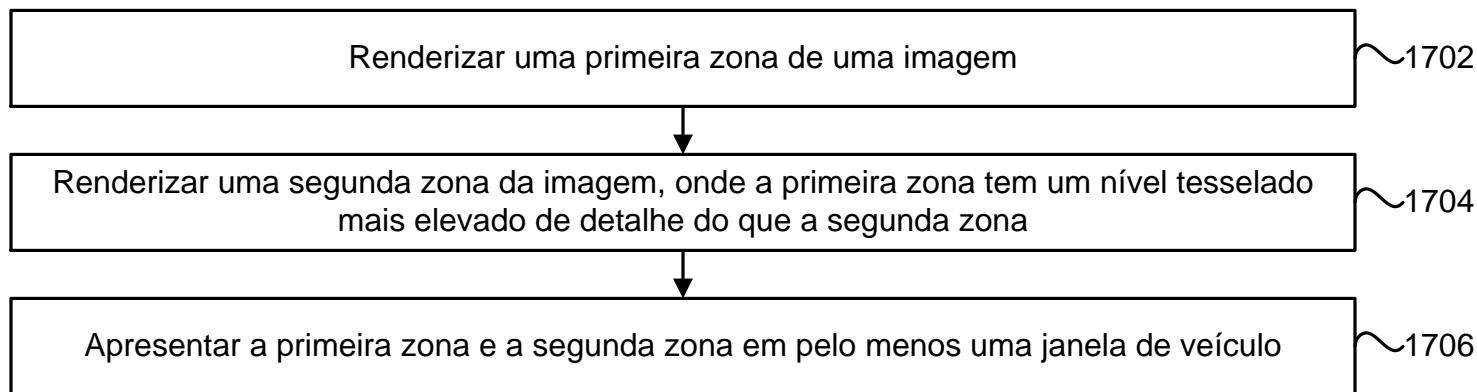


**FIG. 15**



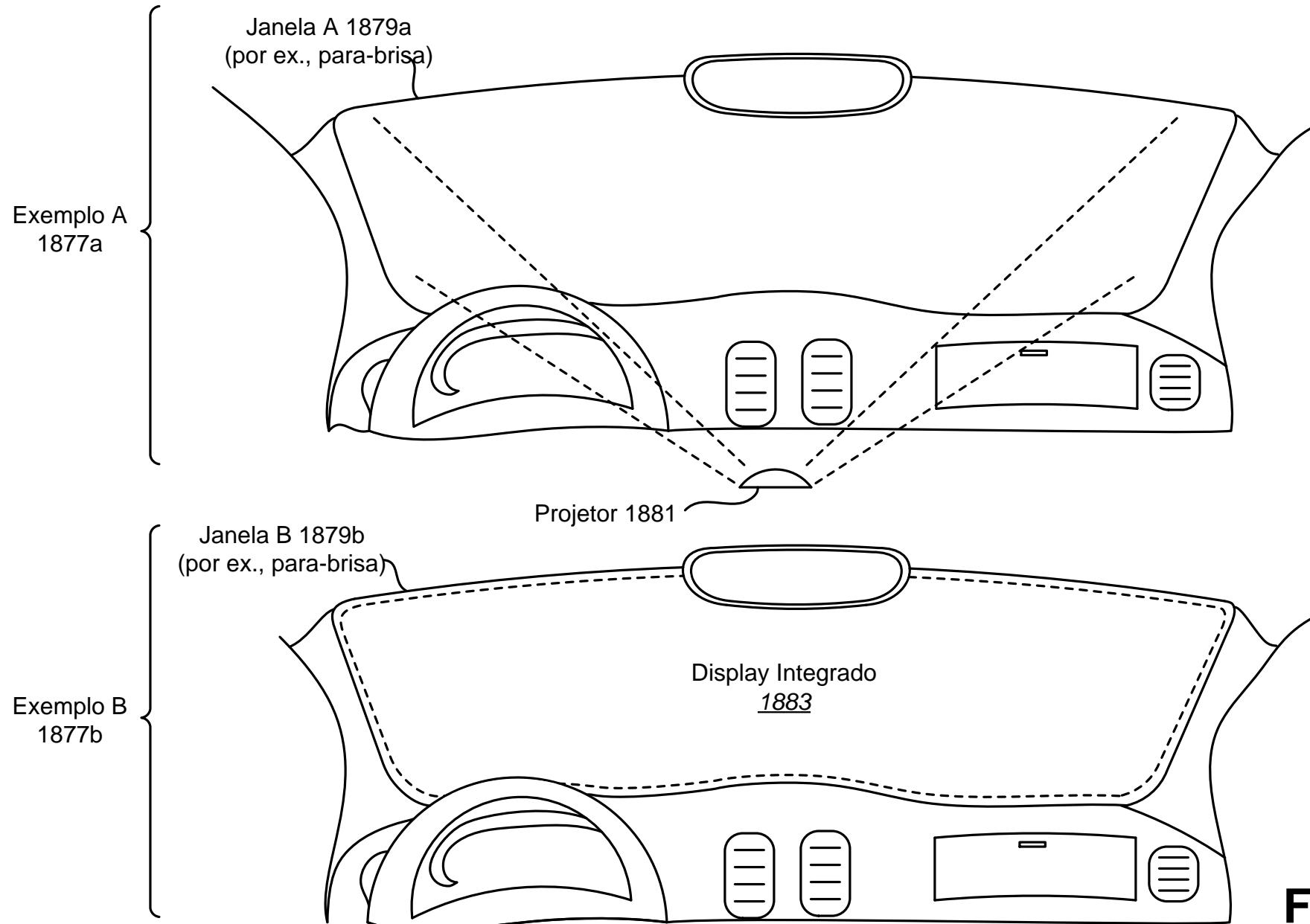
**FIG. 16**

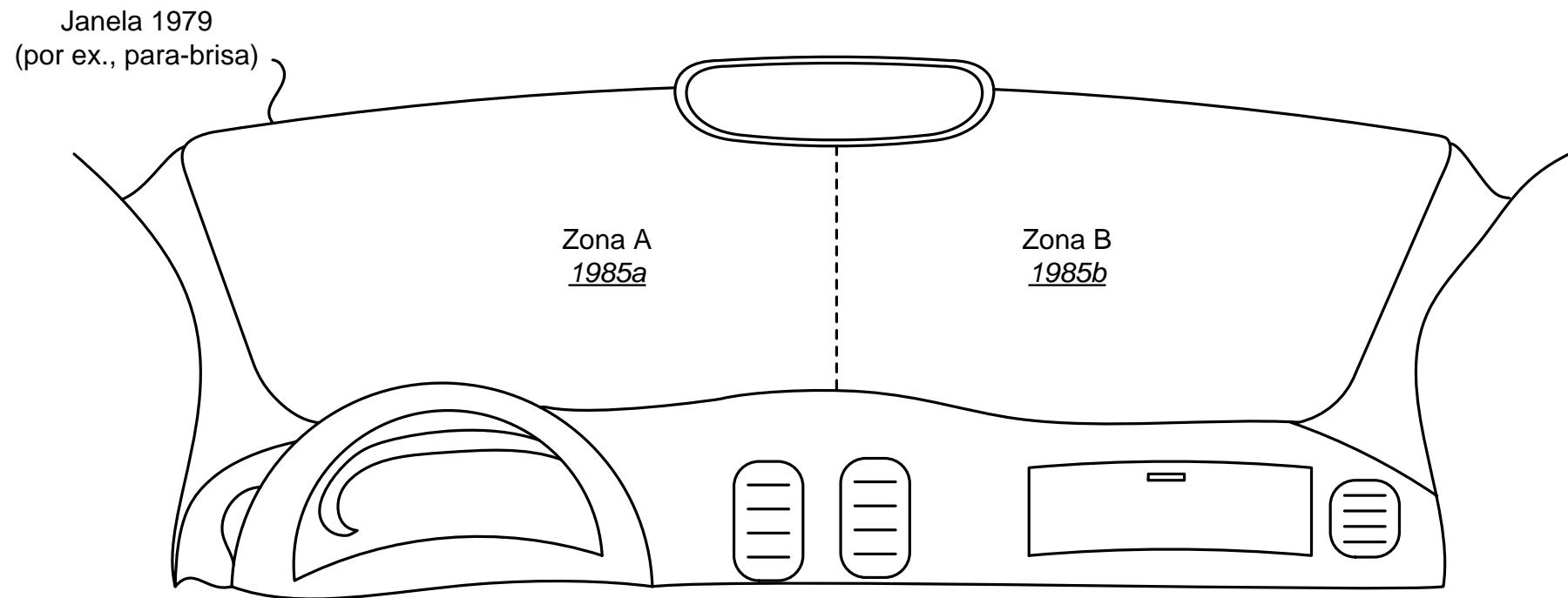
1700 ↘



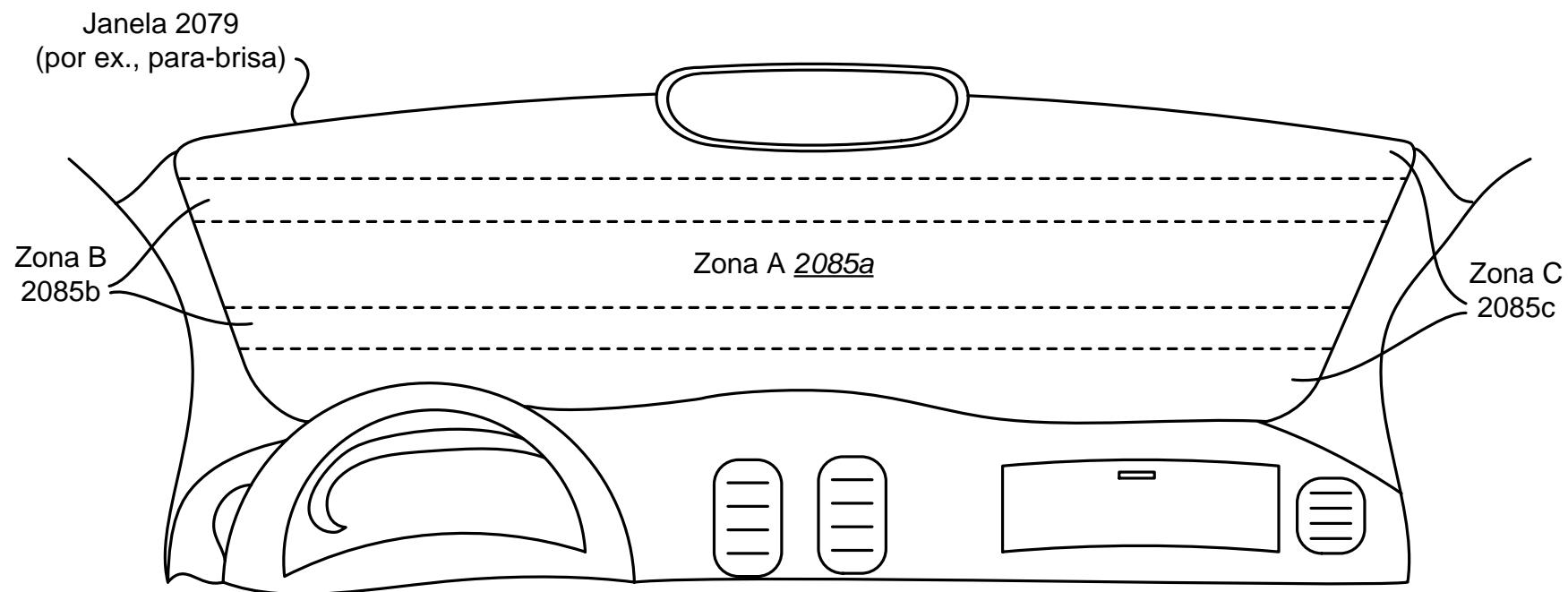
17/24

**FIG. 17**

**FIG. 18**

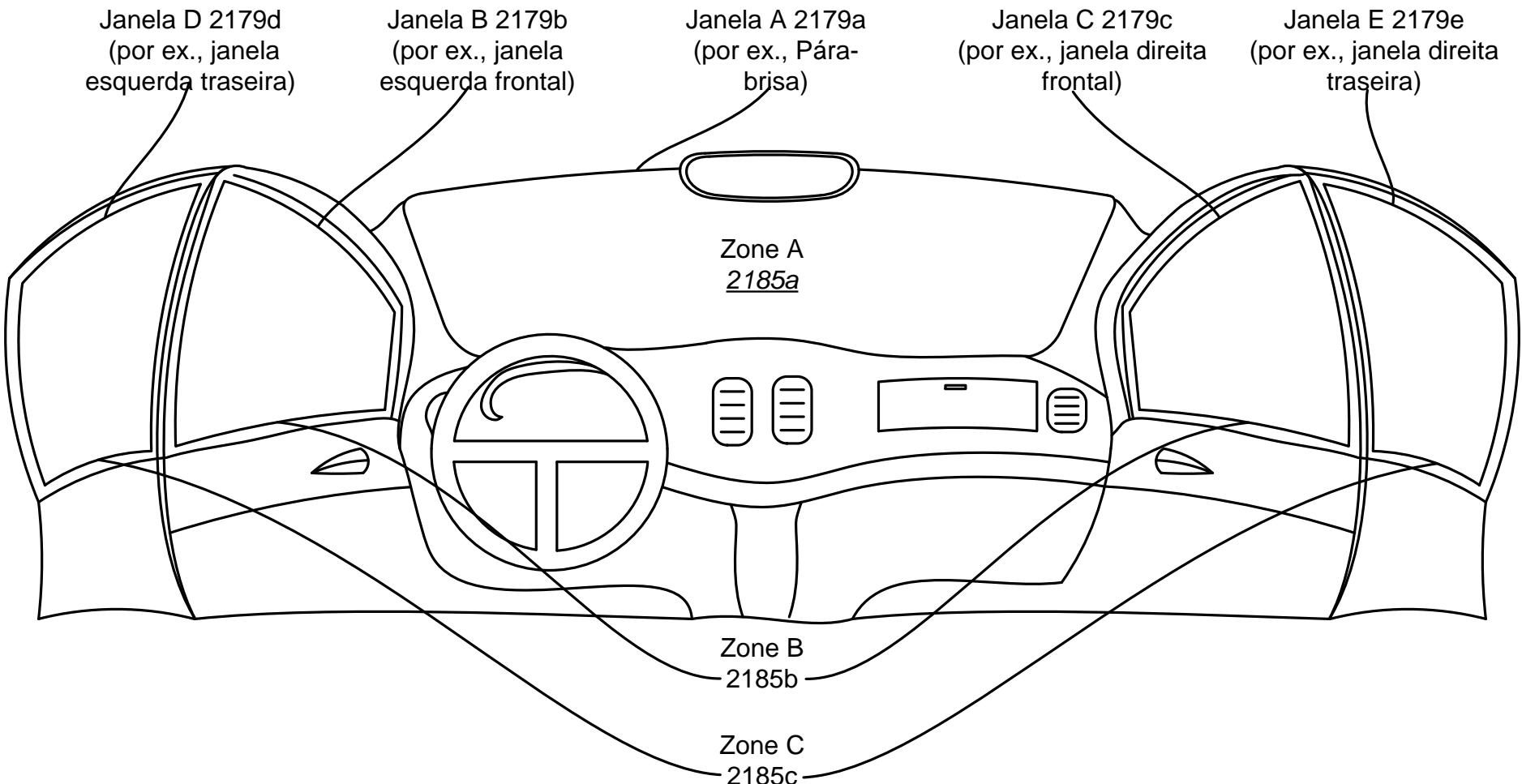


**FIG. 19**



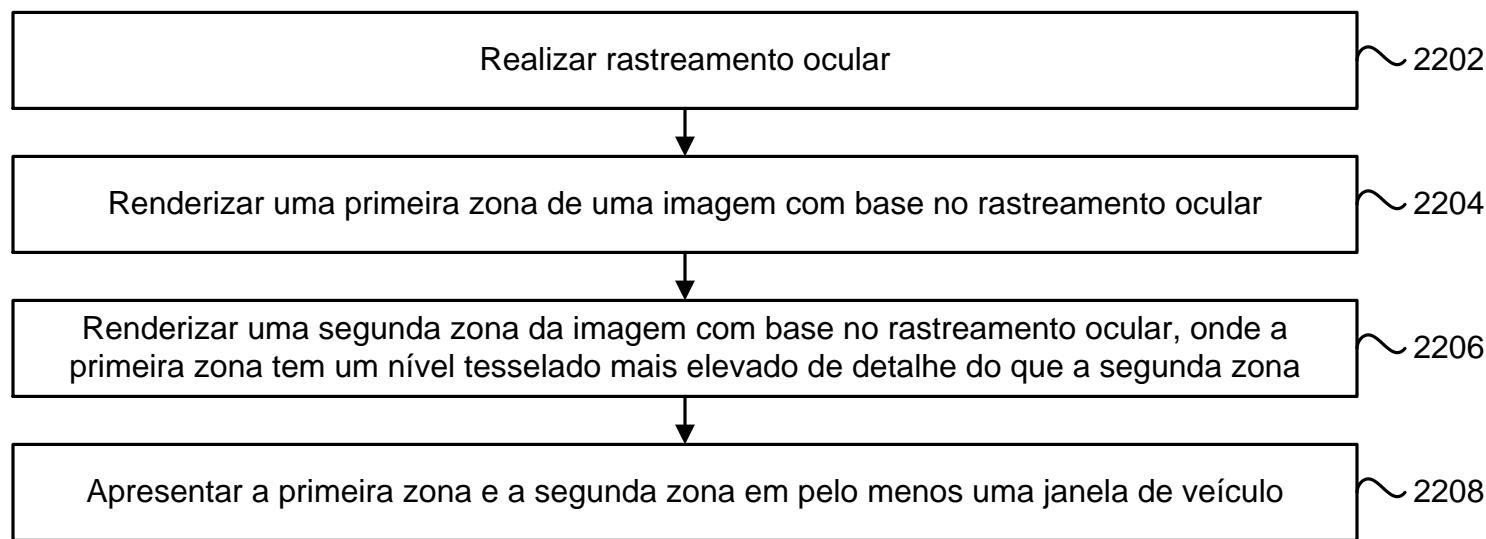
20 / 24

**FIG. 20**



**FIG. 21**

2200 ↘



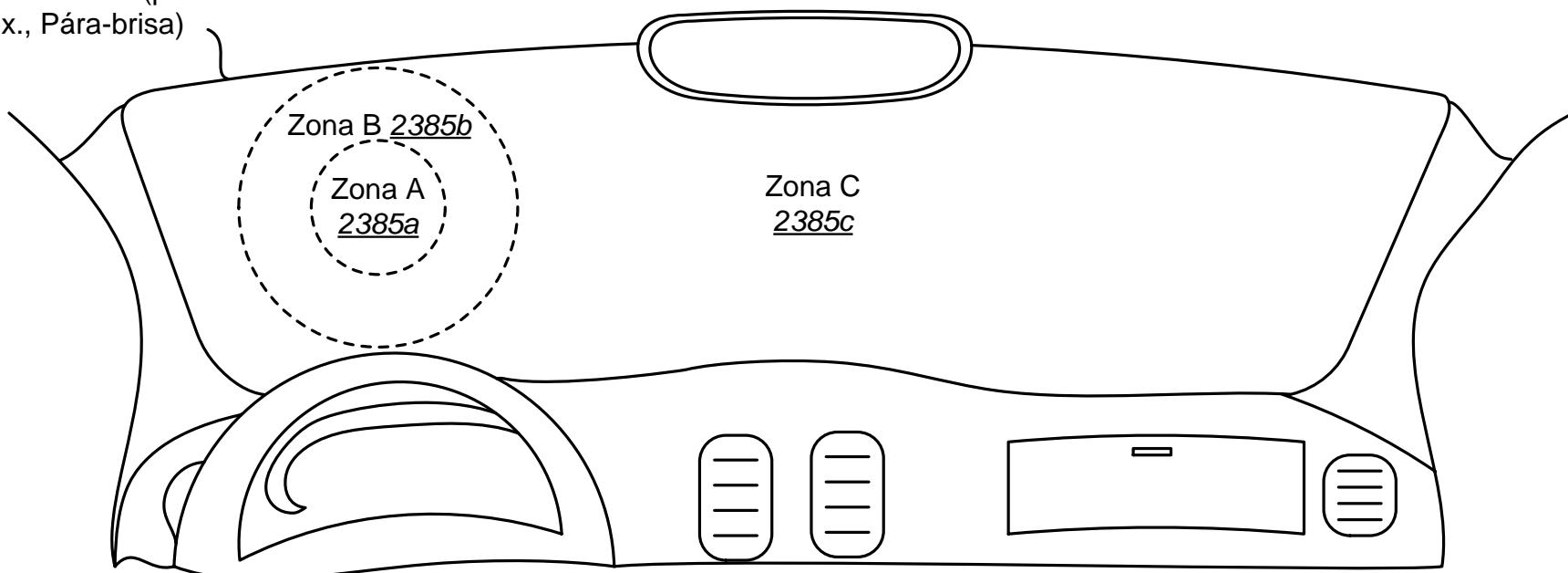
22/24

**FIG. 22**

Janela 2379 (por  
ex., Pára-brisa)

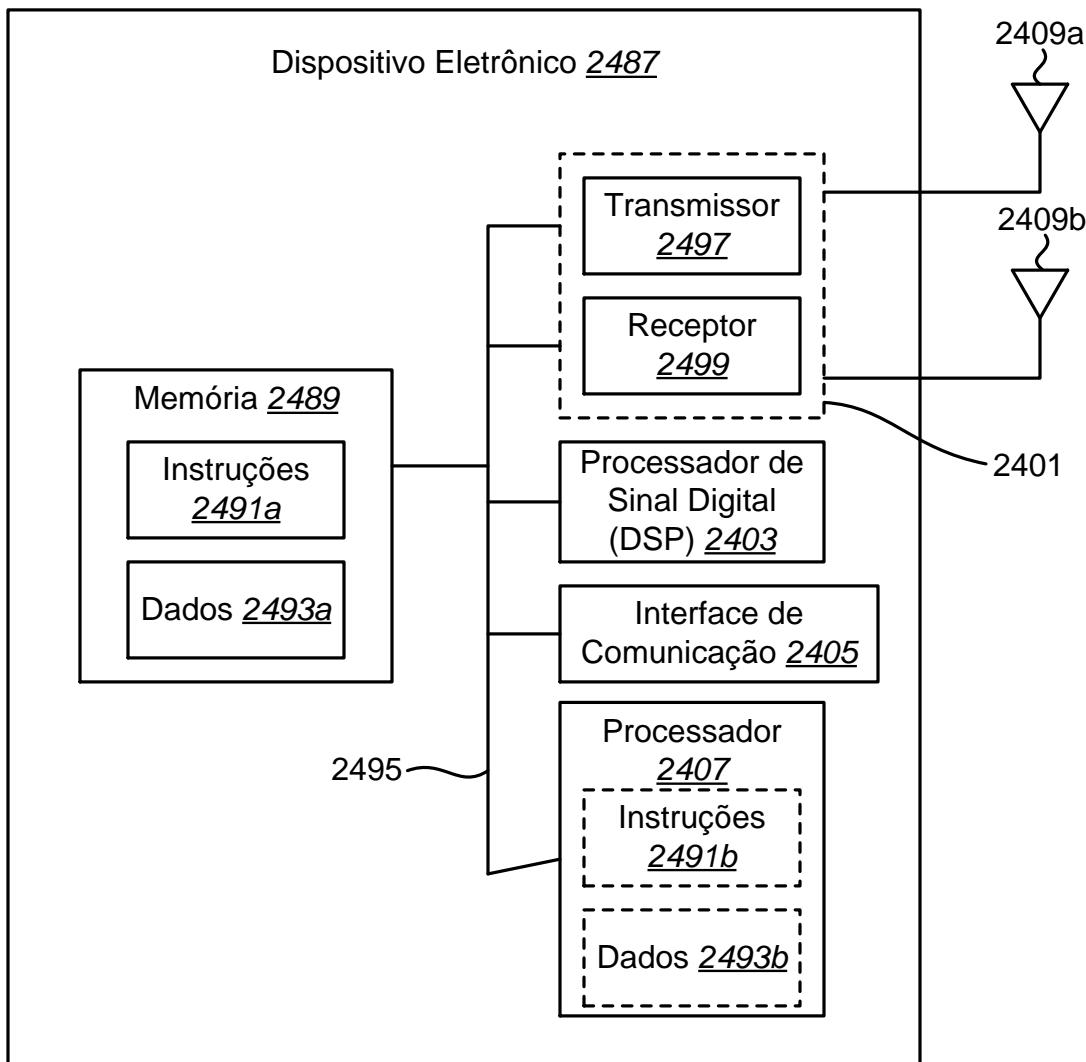
Zona B 2385b  
Zona A  
2385a

Zona C  
2385c



23 / 24

**FIG. 23**



24/24

**FIG. 24**