



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

**(11) PI 0305439-0 B1**

**(22) Data do Depósito:** 14/04/2003

**(45) Data de Concessão:** 31/05/2016



---

**(54) Título:** MÉTODO E SISTEMA PARA FACILITAR INTERAÇÃO ENTRE UM RENDERIZADOR DE VÍDEO E UM ACIONADOR DE DISPOSITIVO GRÁFICO E MEIO ELETRONICAMENTE ACESSÍVEL

**(51) Int.Cl.:** G06T 15/00

**(30) Prioridade Unionista:** 15/04/2002 US 60/372.880, 24/09/2002 US 60/413.060

**(73) Titular(es):** MICROSOFT TECHNOLOGY LICENSING, LLC

**(72) Inventor(es):** STEPHEN J. ESTROP

**"MÉTODO E SISTEMA PARA FACILITAR INTERAÇÃO ENTRE UM RENDERIZADOR DE VÍDEO E UM ACIONADOR DE DISPOSITIVO GRÁFICO E MEIO ELETRONICAMENTE ACESSÍVEL".**

APLICAÇÕES DE PATENTES RELACIONADAS

5 Esta Aplicação, não Provisória para Carta de Patente Norte Americana reivindica o benefício de prioridade de, e neste caso incorpora à guisa de referência a totalidade da divulgação da Aplicação de Patente Norte Americana Provisória, Co-pendente, para Carta de Patente Nº de série 60/413.060, registrada em 24  
10 de setembro de 2002 e intitulada "Métodos para Hardware de Aceleração de Ajustes do tipo 'ProcAmp' de Imagens de Vídeo em uma Exposição de Vídeo" - "Methods for hardware Accelerating the 'ProcAmp' Adjustments of Video on a Computer Display".

Esta Aplicação não Provisória para Carta de pa-  
15 tente também reivindica o benefício de prioridade de e, neste caso incorpora à guisa de referência, em sua totalidade divulgação da Aplicação de Patente Norte Americana Provisória, Co-pendente, para Carta de Patente Nº de série 60/372.880, registrada em 15 de abril de 2002 e intitulada  
20 "Métodos e Dispositivos para Facilitação de Desinterlace de imagens de Vídeo" - "Methods and Apparatuses for Facilitating De-Interlacing of Video Images".

Esta Aplicação não Provisória para Carta de Pa-  
tente Norte Americana refere se à matéria objeto da Aplica-  
25 ção, não Provisória para Carta de Patente Norte Americana Nº de série 10/273.505, registrada em 18 de outubro de 2002 e intitulada "Métodos e Dispositivos para Facilitação de Processamento de Imagens de Vídeo Interlaçadas para Exibições

Esta descrição refere-se, de forma geral, a processamento de dados em imagens /gráficos para exibição e, em particular, à guisa de exemplo mas não limitante, para facilitação de interação entre renderizadores ou representações  
5 de vídeo e acionadores de dispositivos gráficos, que utilizam um protocolo para comunicação de informação entre os mesmos, bem como sua funcionalidade decorrente. Informações deste tipo podem incluir, questionamentos, respostas instruções, etc. que se dirigem a, exemplo, operações de ajuste do  
10 tipo ProcAmp.

#### ANTECEDENTES

Em um ambiente de computador típico, uma placa gráfica ou similar é responsável pela transferência de imagens para um dispositivo de exibição e pela manipulação de  
15 pelo menos parte do processamento das imagens. Para imagens de vídeo, um dispositivo e uma técnica de sobreposição de gráficos é, muitas das vezes, empregada pela placa gráfica e pela totalidade do dispositivo de computação. Por exemplo, para exibição de imagens de vídeo a partir de um DVD ou fonte de feixes de Internet, um procedimento de sobreposição de  
20 gráficos tem início para fins de posicionamento e manutenção das imagens de vídeo.

Um procedimento de sobreposição de gráficos seleciona um retângulo e uma tecla ou encaixe de cor para estabelecimento de localização de tela, na qual a imagem deve  
25 ser exibida. O retângulo pode ser definido com uma coordenada de partida para uma quina do retângulo, em conjunto com a altura e largura desejadas. A cor chave é, geralmente, uma

cor raramente visualizada, tal como rosa brilhante, sendo utilizada para assegurar que o vídeo fica sobreposto no interior do retângulo definido, apenas em caso do vídeo estar logicamente posicionado em uma camada superior de topo de uma área de trabalho na tela de exibição.

Em operação, à medida em que a placa gráfica provê as cores em pixels a um dispositivo de exibição, este efetua um exame para determinar se uma dada localização de pixel se encontra dentro o retângulo gráfico de sobreposição, selecionado. Em caso negativo, o dado de imagem com defeito é conduzido à frente para um dispositivo de exibição. Caso, por outro lado, a dada localização de pixel encontrar-se dentro o retângulo gráfico de sobreposição, selecionado, a placa gráfica efetua um exame para determinar se a imagem com padrão naquele pixel é igual à tecla de cor selecionada. Em caso negativo, o dado de imagem com padrão é conduzido à frente para o dispositivo de exibição para um pixel específico. Caso, por outro lado, a cor do pixel específico seja a tecla de cor selecionada, a placa gráfica adianta os dados de imagem de vídeo ao dispositivo de exibição para aquele dado pixel.

Existem, infelizmente, diversas desvantagens em relação a esta técnica de sobreposição gráfica. Em primeiro lugar, existem, de forma geral, apenas recursos de hardware suficientes, para efetuar um único procedimento de técnica de sobreposição gráfica a cada vez. Independentemente, o baseamento na técnica de sobreposição gráfica sempre resulta em restrições na quantidade de exibições possíveis de vídeo

simultâneas, limitadas pelo hardware. Em segundo lugar, rosa ou outra tecla de cor, algumas das vezes se torna visível (isto é, exibida em um dispositivo de exibição associado), quando a janela contendo o vídeo exibido é movida vigorosamente em torno da área de trabalho na tela de exibição.

Em terceiro lugar, um comando de impressão de tela não funciona efetivamente, visto que a imagem de vídeo que é exibida no dispositivo de exibição não é capturada pelo comando de impressão de tela. Ao invés disso, a tecla de cor é capturada pelo comando de impressão de tela, desta forma a imagem impressa (ou copiada e arquivada) inclui um retângulo sólido da tecla de cor, no local em que o vídeo é exibido no dispositivo de exibição.

Outra técnica para exibição de imagens de vídeo se baseia no uso de um microprocessador hospedeiro para execução de ajustes de vídeo anteriormente à transferência da imagem de vídeo ao processador gráfico para adiantamento, ao dispositivo de exibição. Existem também diversas desvantagens em relação a esta técnica de processador hospedeiro. Em primeiro lugar, o microprocessador hospedeiro e o sub-sistema de memória associado de um ambiente de computação típico não se encontram otimizados para processamento de amplas imagens de vídeo. Conseqüentemente, a dimensão e a quantidade das imagens de vídeo que podem ser exibidas são severamente restritas, em segundo lugar, para que o microprocessador hospedeiro trabalhe efetivamente, a imagem de vídeo deve residir na memória que pode ser diretamente endereçada pelo microprocessador hospedeiro. Como resultado, outros tipos de

aceleração de hardware, tais como descompressão e/ou desinterlace não podem ser efetuados na imagem e vídeo.

Em resumo, as técnicas anteriores tais como procedimento de sobreposição gráfica e com base no processador hospedeiro, que resultam em artefatos visuais, são por demais lentas e/ou fazem uso de recursos de memória insuficientes, sendo limitadas em relação ao hardware, restringem a flexibilidade de apresentação no vídeo e/ou não possibilitam um comando de impressão de tela completamente funcional. Assim sendo, existe a necessidade de esquemas e/ou soluções para remediação destas e outras deficiências por meio de, *inter alia*, facilitação a interação entre renderizadores ou representações de vídeo e acionadores de dispositivos gráficos.

#### 15                    RESUMO

A facilitação de interação entre representações de vídeo e acionadores de dispositivos gráficos pode ser possibilitada através de protocolos de comunicação e/ou interfaces de aplicação de programação (APIs), que permitem informações que se referem a capacidades de processamento de imagem de hardware de gráficos associados, a serem trocados entre um acionador de dispositivo gráfico e um renderizador ou uma representação de vídeo. As capacidades de processamento de imagens incluem capacidades de processamento de vídeo; estas capacidades de processamento de vídeo incluindo, à guisa de exemplo, mas não de forma limitante, ajustes de controle de amplificador de processo (ProcAmp), desinterlace, correções de proporções de aspecto, conversões de espaço

de cor, conversões de proporção de estrutura, espelhamento vertical e horizontal e misturas alfa.

Em um exemplo de implementação de método, um método facilita a interação entre um ou mais renderizadores e, pelo menos um acionador de dispositivo gráfico, o método incluindo ações de: consulta, por meio de um renderizador de vídeo, dentre um ou mais renderizadores de vídeo, o pelo menos um acionador de dispositivo gráfico relacionado a capacidades de processamento de vídeo e informações, por meio do pelo menos um acionador de dispositivo gráfico, ao renderizador de vídeo de pelo menos um subconjunto de capacidades de processamento de vídeo, que o pelo menos um acionador de dispositivo gráfico pode oferecer o renderizador de vídeo.

Em um primeiro exemplo de meio de implementação, instruções executáveis eletronicamente do mesmo para um renderizador de vídeo precipita ações incluindo: emissão de uma consulta a partir do renderizador de vídeo em direção ao acionador de dispositivo gráfico, a consulta requisitando informações referentes às capacidades ProcAmp e recebimento uma resposta no renderizador de vídeo, a partir do acionador de dispositivo gráfico, esta resposta incluindo a informação requisitada referente às capacidades ProcAmp.

Em um segundo exemplo de meio de implementação, instruções executáveis eletronicamente do mesmo, para um acionador de dispositivo gráfico, precipita ações incluindo: recebimento de uma consulta em um acionador de dispositivo gráfico a partir do renderizador de vídeo, esta consulta requisitando informações referentes às capacidades ProcAmp e

envio de uma resposta ao renderizador de vídeo, a partir do acionador de dispositivo gráfico, esta resposta incluindo a informação requisitada que se refere às capacidades ProcAmp.

Em um exemplo de sistema de implementação, um sistema facilita a interação entre um renderizador de vídeo e um acionador de dispositivo gráfico, o sistema incluindo: uma lógica de renderização de vídeo que se adapta ao preparo de consultas que requisitem informações referentes às capacidades ProcAmp, que possam ser aplicadas ao vídeo, que está para ser exibido e uma lógica de acionador de dispositivo gráfico que se adapte ao preparo de respostas que indiquem quais capacidades ProcAmp podem ser aplicadas ao vídeo que está para ser exibido.

Outros métodos, sistemas, dispositivos, protocolos, meios, disposições, etc., implementações encontram-se descritos neste caso.

#### BREVE DESCRIÇÃO DOS DESENHOS

Os mesmos numerais são utilizados através dos desenhos para fazer referência a aspectos, características e componentes similares e/ou correspondentes.

A Figura 1 é uma encadeamento de processamento de vídeo, que inclui operação de ajuste ProcAmp.

A Figura 2 é uma segunda encadeamento que inclui duas operações de processamento de vídeo, para alcance de um alvo de renderização RGB.

A Figura 3 é uma terceira encadeamento de processamento de vídeo, que inclui uma operação de processamento de vídeo, para alcance em um alvo de renderização RGB.

A Figura 4 é um diagrama em blocos, que ilustra determinados elementos funcionais de um dispositivo de computação ou outro dispositivo eletrônico, que se encontra configurado para facilitar a interação entre renderizadores de vídeo e acionadores de dispositivos gráficos.

A Figura 5 é um diagrama de comunicação / sinalização, que ilustra um exemplo de protocolo entre um renderizador de vídeo e um acionador de dispositivo gráfico.

A Figura 6 é um diagrama de fluxo que ilustra um exemplo de método para facilitar a interação entre um renderizador de vídeo e um acionador de dispositivo gráfico.

A Figura 7 ilustra um exemplo de ambiente de operação de computação (ou dispositivo eletrônico, em geral), que é capaz de implementar (total ou parcialmente), pelo menos um aspecto para facilitar a interação entre renderizadores de vídeo e acionadores de dispositivos gráficos, conforme descrito neste caso.

#### DESCRIÇÃO DETALHADA

Exemplos de Encadeamentos de Processamento de Vídeo e Ajustes ProcAmp

Exemplo de Encadeamento de Processamento de Vídeo com Ajuste ProcAmp

A Figura 1 é uma primeira encadeamento de processamento de vídeo 100, que inclui uma operação de ajuste ProcAmp 104. A primeira encadeamento de processamento de vídeo 100 pode ser implementada com a utilização de um hardware gráfico, tal como uma placa gráfica. Este inclui (i) três blocos de memória de imagem 102, 106 e 108 e (ii) pelo menos

uma operação de processamento de imagem 104. O bloco de memória de imagem 102 inclui uma superfície plana fora de tela de imagem de vídeo YUV. A operação de processamento de imagem 104, a qual compreende uma operação de ajuste ProcAmp 5 104, conforme ilustrada, é aplicada ao bloco de memória de imagem 102 para fins de produção do bloco de memória de imagem 106. O bloco de imagem de memória 106 inclui uma superfície plana fora de tela YUV ou uma textura YUV, dependendo dos parâmetros e capacidades do hardware gráfico, que se encontra efetuando as operações de ajuste de imagem. 10

Após uma ou mais operações de processamento de imagem adicionais (não explicitamente ilustradas na Figura 1), o hardware gráfico produz um bloco de memória de imagem 108, que inclui um alvo de renderização RGB. Este alvo de 15 renderização RGB de bloco de memória de imagem 108 pode ser exibido em um dispositivo de exibição por meio do hardware gráfico, sem operações adicionais de processamento de imagem. Do mesmo modo, o bloco de memória de imagem 108 inclui dados de imagem para cada pixel de tela de um dispositivo de 20 exibição, de tal forma que nenhum dado de imagem necessite ser recuperado a partir de outra memória, durante a adiantamento dos dados de imagem provenientes do bloco de memória de imagem 108, até o dispositivo de exibição.

A operação de ajuste ProcAmp 104, se refere a um 25 ou mais ajustes de amplificador de processo (ProcAmp). O conceito de ajustes ProcAmp, foi originado quando vídeos foram armazenados, manipulados e exibidos com a utilização de técnicas análogas. No entanto, operações de ajuste ProcAmp

104 podem ser agora efetuadas com a utilização de técnicas digitais. Tais operações de ajuste ProcAmp 104 podem incluir uma ou mais operações, que se direcionem a um ou mais dentre pelo menos as seguintes propriedades de vídeo: brilho, contraste, saturação e matiz.

#### Exemplo de Propriedades de Vídeo Relacionadas a ProcAmp

As seguintes descrições de brilho, contraste, saturação e matiz, em conjunto com ajustes possíveis e/ou sugeridos para manipulação de seus valores, tratam-se de um exemplo de descrição de implementação. Outras linhas de condução para ajustes ProcAmp podem ser alternativamente empregadas.

Brilho: o brilho é alternativamente conhecido como "ajuste escuro"; o brilho não devendo ser confundido com ganho (contraste). Esse é utilizado para ajustar o nível de "visualização escura" em cada cenário específico de visualização. Funcionalmente, este adiciona ou subtrai a mesma quantidade de etapas (bits) de quantificação a partir de todas as palavras de luminescência em um quadro de imagem. O brilho pode e, de forma geral,, cria situações de grampeamento, caso o deslocamento, adicionado à palavra de luminescência seja 0 ou maior do que uma proporção total. Este, de forma geral, é interativo com o controle de contraste.

Contraste: o contraste trata-se de um "ganho" da luminescência na imagem. Este é utilizado para alterar a luz relativa aos valores escuros em um quadro de imagem. Funcionalmente, este é um ganho linear, positivo ou negativo que

mapeia o alcance de chegada de valores em uma proporção maior ou menor. O ponto de ajuste (por exemplo, nenhuma modificação como modificação de ganho), normalmente é igual a um código 0, mas sendo mais apropriado à palavra de código que se encontre associada a um ponto de ajuste nominal de visualização escuro (preto). A estrutura de ganho de contraste é, de forma geral, uma rampa de transferência linear, que passa através do ponto de ajuste. Funções de contraste, geralmente, envolvem arredondamento de valores computados, caso o ganho tenha sido ajustado a algum valor diferente de 1 a -1, e este arredondamento, de forma geral, inclui um tremor de programação para evitar a geração de "contorno" do artefato visível.

Saturação: a saturação é o equivalente lógico ao contraste. Trata-se de uma função de ganho, com um ponto de ajuste em torno de "chroma zero" (por exemplo, código 128 em YUV ou código 0 em RGB, em uma implementação descrita).

Matiz: o matiz é o relacionamento de fase de componentes de cromatização. Matiz é tipicamente especificado em graus, com uma faixa válida de - 180 até + 180 e um padrão de 0 graus. O matiz em um sistema componente (por exemplo YUV ou RGB) é uma variável em três partes, na qual os três componentes se modificam, em conjunto, a fim de manter relações cromatográficas / de luminescência válidas.

#### Exemplo de Ajuste Relacionado a ProcAmp no Espaço de Cor YUV

As seguintes descrições para processamento de brilho, contraste, saturação e matiz em um espaço de cor YUV,

em conjunto com ajustes possíveis e/o sugeridos para manipulação de seus valores, tratam-se de um exemplo para a implementação descrita. Outras linhas de condução de ajustes podem ser alternativamente empregadas. De forma geral, trabalhando-se em um espaço de cor YUV, simplifica os cálculos que são envolvidos para controle de ajuste ProcAmp de um feixe de vídeo.

Processamento Y: subtrai-se dezesseis (16) dos valores Y para posicionar o nível escuro (preto) em zero. Isso remove o deslocamento DC, de forma que o ajuste de contraste não varie o nível preto. Devido ao fato dos valores Y serem menores do que 16, valores Y negativos deveriam ser apoiados nesse ponto de processamento. O contraste é ajustado por meio da multiplicação dos valores de pixel YUV por um contraste. (caso U e V estejam ajustados, uma comutação de cor irá resultar, caso o contraste seja modificado). O valor de propriedade de brilho é adicionado (ou subtraído) dos valores de contraste ajustados Y; isso impedindo que um deslocamento DC seja introduzido devido ao ajuste de contraste. Finalmente, o valor 16 é adicionado novamente para reposição do nível preto a 16. Um exemplo de equação para o processamento dos valores Y então é:

$$Y' = ((Y - 16) \times C) + B + 16,$$

onde C é o valor de Contraste e B é o valor do Brilho.

Processamento UV: em primeiro lugar subtrai-se cento e vinte e oito (128) tanto dos valores U, quanto dos valores V para posicionar a faixa em torno de zero. A pro-

priedade de matiz, isolada, é implementada por meio da mistura dos valores U e V, em conjunto como se segue:

$$U' = (U - 128) \times \cos (H) + (V - 128) \times \sin (H), \text{ e}$$

$$V' = (V - 128) \times \cos (H) - (U - 128) \times \sin (H),$$

5 onde H representa a ângulo de matiz desejado.

A saturação é ajustada por meio da multiplicação, tanto de U, quanto de V por um contraste, em conjunto ao valor de saturação. Finalmente, o valor 128 é adicionado de volta, tanto a U, quanto a V. O processamento combinado de Matiz e Saturação nos dados UV, é portanto:

$$U' = ((U - 128) \times \cos (H) + (V - 128) \times \sin (H)) \times C \times S) + 128, \text{ e}$$

$$V' = ((V - 128) \times \cos (H) - (U - 128) \times \sin (H)) \times C \times S) + 128,$$

15 onde C trata-se do valor de Contraste como na equação Y', acima e H é o ângulo de Matiz e S é a Saturação.

#### Exemplo de Encadeamento de Processamento de Vídeo com Duas Operações de Processamento

A Figura 2 trata-se de uma segunda encadeamento processamento de vídeo 200, que inclui duas operações de processamento de vídeo 202 2 206 para alcance em um alvo de renderização RGB 108. A segunda encadeamento de processamento de vídeo inclui (i) três blocos de memória de imagem 102, 204 e 108 e (ii) duas operações de processamento de imagem 25 202 e 206.

Para uma segunda encadeamento de processamento de vídeo 200, de forma geral, o bloco de memória de imagem 204 inclui uma textura RGB. O bloco de memória de imagem RGB 204

resulta do bloco de memória de imagem 102, após a aplicação da operação de processamento de imagem 202. O bloco de memória de imagem 108 é produzido a partir do bloco de memória de imagem 204, após a operação de processamento de imagem  
5 206.

Outras operações de processamento de imagem, em adição ao ajuste de controle ProcAmp, podem ser efetuadas. Por exemplo, qualquer uma ou mais de uma dentre as operações de processamento de vídeo podem ser aplicadas aos dados de  
10 imagem de vídeo, anteriormente a sua exibição em uma tela de um dispositivo de exibição:

- 1.ajustes de controle ProcAmp
- 2.desinterlace
- 3.correção de proporção de aspecto
- 15 4.conversão de espaço de cor, e
- 5.espelhamento vertical ou horizontal e misturas alfa.

Quando possível, as operações de processamento de vídeo desejadas (e/ou outras imagens) são combinadas em tão  
20 poucas operações quanto possíveis, de forma a reduzir a amplitude de memória total que é consumida, durante o processamento das imagens de vídeo. O grau, com o qual as operações de processamento podem ser combinadas é, de forma geral, determinado através das capacidades do hardware gráfi-  
25 co. Tipicamente, um processamento de conversão de espaço de cor e um processamento de correção de proporção de aspecto são aplicados a muitos, se não a todos, os feixes de vídeo.

No entanto, o espelhamento vertical / horizontal e as misturas alfa são aplicadas com menos frequência.

Para uma segunda encadeamento de processamento de vídeo 200, são combinados o processamento de ajuste ProcAmp e o processamento de conversão de espaço de cor em uma operação de processamento de imagem 202. O processamento de correção de proporção de aspecto é efetuado com a operação de processamento de imagem 206. Opcionalmente, o espelhamento e/ou a mistura alfa podem ser combinados em uma operação de processamento de imagem 206. Como ilustrado, o hardware gráfico que se encontra implementando a segunda encadeamento de processamento de vídeo 200, faz uso de duas operações de processamento de imagens e de três blocos de memória de imagem, para produção do bloco de memória de imagem 108, como alvo de renderização RGB. No entanto, alguns hardware gráficos podem ser mais eficientes.

#### Encadeamento de Processamento de Vídeo com uma Operação de Processamento

A Figura 3 trata-se de uma terceira encadeamento de processamento de vídeo 300, que inclui uma operação de processamento de vídeo 302, para alcance do alvo de renderização RGB 108. De forma geral, a terceira encadeamento de processamento de vídeo 300 é implementada com hardware gráfico, fazendo uso de uma operação de processamento de imagem 302 e dois blocos de memória de imagem 102 e 108. Especificamente, o bloco de memória de imagem 108 é produzido a partir do bloco de memória de imagem 102, através da operação de processamento de imagem 302. A operação de processamento

de imagem 302, como ilustrada, inclui múltiplas operações de processamento de vídeo, como descritas abaixo.

A terceira encadeamento de processamento de vídeo 300 é mais curta do que a segunda encadeamento de processamento de vídeo 200 (da Figura 2), devido à operação de processamento de imagem 302 combinar o processamento de ajuste ProcAmp, o processamento de conversão de espaço de cor e o processamento de correção de proporção de aspecto. A quantidade de estágios em uma dada encadeamento de processamento de vídeo, portanto, depende da quantidade e dos tipos de operações de processamento de imagens que são requisitadas pelo software (por exemplo, uma aplicação, um componente de sistema operacional, etc.), exibindo a imagem de vídeo, bem como as capacidades do hardware gráfico associado. Exemplos de software, hardware gráfico e assim por diante se encontram adicionalmente descritos abaixo, com referência à Figura 4.

#### Exemplo de Software Relacionado a Vídeo e Hardware Gráfico

A Figura 4 é um diagrama em bloco 400, que ilustra determinados elementos funcionais de um dispositivo de computação ou outro dispositivo eletrônico, que se encontra configurado para facilitar a interação entre um renderizador de vídeo 410 e um acionador de dispositivo gráfico 422. Estes vários exemplos de elementos e/ou funções podem ser implementados em hardware, software, firmware, algumas misturas destes e assim por diante. Tais hardwares, softwares, firmwares, algumas misturas destes e assim por diante são refe-

ridos neste caso, em conjunto e separadamente, genericamente como lógica.

A configuração do diagrama em bloco 400 trata-se apenas de um exemplo de dispositivo ou sistema de processamento de dados de vídeo. Deve ser entendido que um ou mais 5 dentre os elementos e/ou funções ilustrados e descritos podem ser combinados, rearranjados, ampliados, omitidos, etc., sem invalidar uma capacidade de facilitação a interação entre renderizadores de vídeo e acionadores de dispositivos 10 gráficos.

Dispositivos ou sistemas 400 incluem uma lógica de transformação 408, a qual, por exemplo, pode incluir instruções efetuadas por uma unidade de processamento central (CPU), uma unidade de processamento gráfico e/ou misturas 15 das mesmas. A lógica de transformação 408 encontra-se configurada para receber dados de vídeo codificados a partir de pelo menos uma fonte 406. Os dados de vídeo codificados provenientes de uma fonte 406 são de alguma forma codificados (por exemplo, MPEG-2, etc.) e a lógica de transformação 408 20 encontra-se configurada para decodificar estes dados de vídeo codificados.

À guisa de exemplo, a fonte 406 pode incluir um disco magnético e um acionamento (drive) de disco relacionado, um disco ótico e um acionador de disco relacionado, uma 25 fita magnética e um acionador de fita relacionado, memória em estado sólido, sinal transmitido, meio de transmissão ou outra fonte similar, configurada para oferecer ou prover de alguma forma os dados de vídeo codificados à lógica de

transformação 408. Exemplos adicionais de fonte 406 se encontram descritos abaixo, com referência à Figura 7. Em determinadas implementações, a fonte 406 pode incluir múltiplos componentes de fonte, tais como uma fonte em rede e uma  
5 fonte remota. Conforme ilustrado, a fonte 406 inclui Internet 404 e armazenamento remoto com base em disco 402.

Os dados de vídeo decodificados que são fornecidos pela lógica de transformação 408 são providos a pelo menos um renderizador de vídeo 410. À guisa de exemplo, mas  
10 não de limitação, o renderizador de vídeo 410 pode ser concretizado com a utilização de um Misturador de Renderizador de Vídeo - Video Mixer and Renderer - (VMR) de um sistema operacional (OS) Microsoft® Windows®. Em uma implementação descrita, o renderizador de vídeo 410 encontra-se configura-  
15 do para auxiliar a lógica de transformação 408 na decodificação do feixe de vídeo e para levar as operações de processamento de vídeo a serem executadas, para combinar quaisquer outros dados de imagem auxiliares, tais como Captação fechada - Closed Caption (CCs) ou imagens em sub - quadros de DVD  
20 com a imagem de vídeo e assim por diante. Além disso, no tempo apropriado, o renderizador de vídeo 410 submete ou leva à submissão dos dados de imagem de vídeo à lógica de interface gráfica 412, para eventual exibição em um dispositivo de exibição.

25 Os dados de vídeo renderizados resultantes são desta forma providos a uma lógica de interface gráfica 412. À guisa de exemplo, mas não de limitação, a lógica de interface gráfica 412 pode incluir, por exemplo DirectDraw®, Di-

rect3D<sup>®</sup> e/ou outra lógica similar. A lógica de interface gráfica 412 encontra-se configurada para prover uma interface entre o renderizador de vídeo 410 e um dispositivo gráfico424. Conforme ilustrado, o dispositivo gráfico424 inclui  
5 uma unidade de processamento gráfico(GPU) 426, uma memória de vídeo 432 e um conversor digital para analógico (DAC) 434. À guisa de exemplo, mas não de limitação, o dispositivo gráfico424 pode ser concretizado em forma de uma placa gráfica de vídeo, que se encontra configurada no interior de um  
10 dispositivo de computador ou outro dispositivo eletrônico.

O fornecimento de dados de imagem através da lógica de interface gráfica 412 encontra-se provido em um acionador de dispositivo gráfico422 fazendo uso de uma interface de acionador de dispositivo (DDI) 414. Na Figura 3, a  
15 interface de acionador de dispositivo 414 encontra-se ilustrada como apresentando pelo menos uma interface de aplicação de programa (API) 416, associada à mesma. A interface de acionador de dispositivo 414 encontra-se configurada para suportar e/ou estabelecer a interface dentre o renderizador  
20 de vídeo 410 e o acionador de dispositivo gráfico422.

Como ilustrado no dispositivo / sistema 400 e para a implementação descrita, a interface de acionador de dispositivo 414 e o acionamento de dispositivo gráfico422 podem ainda ser categorizados como constituindo parte de um modo  
25 de usuário 418 ou modo kernel ou de parte central 420, com relação ao ambiente de sistema operacional associado e ao dispositivo gráfico424. Conseqüentemente, o renderizador de vídeo 410 e a interface de vídeo 414 fazem parte do modo de

usuário 418 e o acionador de dispositivo gráfico422 faz parte do modo central 420. Estas comunicações ocorrendo, pelo menos entre a interface de acionador de dispositivo 414 e o acionador de dispositivo gráfico422 se cruzam entre o modo  
5 de usuário 418 e o modo central 420.

Nesta implementação descrita, os dados de imagem de vídeo que são fornecidos pelo renderizador de vídeo 410 são então providos à unidade de processamentográfico426. A unidade de processamentográfico426 pode ser configurada para  
10 efetuar uma ou mais operações de processamento de imagem. Estas operações de processamento de imagem incluem ajustes ProcAmp e/ou outras operações de processamento de vídeo, como indicado pela lógica de ajuste ProcAmp 428 e/ou outra lógica de operação de processamento de vídeo 430, respectiva-  
15 mente. As operações de ajuste ProcAmp e outras operações de processamento de vídeo, tais como de desinterlace e conversão de taxa de quadro se encontram descritas abaixo, bem como acima.

O fornecimento proveniente da unidade de processamentográfico426 é provido à memória de vídeo 432. Quando a  
20 memória de vídeo 432 é lida, os dados resultantes de imagem podem ser conduzidos à frente para um conversor digital para analógico 434, o qual fornece uma sinal de vídeo correspondente que é adequado para exibição no e pelo dispositivo de  
25 exibição 436. Em outras configurações, o dispositivo de exibição 436 pode ser capaz de exibir os dados de margem digitais a partir da memória de vídeo 432, sem conversão analógica, por meio de um conversor digital para analógico 434.

Exemplo de Protocolo entre Renderizador de Vídeo e Acionador de Dispositivo Gráfico

A Figura 5 trata-se de um diagrama de comunicação / sinalização 500, que ilustra um exemplo de protocolo entre um renderizador de vídeo 410 e um acionador de dispositivo gráfico 422. O exemplo de protocolo facilita a realização da operação de processamento de vídeo (ou outra imagem). Tal como um ajuste ProcAmp, esta operação de processamento de vídeo pode incluir aquelas que são requisitadas / especificadas por uma aplicação ativada por usuário e uma exibição de vídeo controlada (por exemplo, uma aplicação de instigação).

O diagrama de comunicação / sinalização 500 inclui múltiplas trocas e transmissões de comunicação entre renderizadores de vídeo 410 e o acionador de dispositivo gráfico 422. Opcionalmente, as comunicações podem ser possibilitadas e/ou auxiliadas por uma interface gráfica 412 (da Figura 4) e/ou interface de acionamento de dispositivo 414, em conjunto com qualquer APIs 416, aplicável, da mesma.

Uma troca de comunicação 502 se direciona ao estabelecimento de capacidades de processamento de vídeo (VP). Especificamente, o renderizador de vídeo 410 requisita ou indaga na transmissão 502A do acionador de dispositivo gráfico 422, em relação às capacidades de processamento de vídeo que são possuídas por e que devem ser providas pelo acionador de dispositivo gráfico 411. Em resposta 502B, o acionador de dispositivo gráfico 422 informa ao renderizador

de vídeo 410 das capacidades de processamento de vídeo alocadas.

As capacidades de processamento de vídeo alocadas incluem aquelas operações de processamento de vídeo que a-  
5 cionador de dispositivo gráfico422 é capaz de efetuar. Estas podem incluir uma ou mais operações de ajuste de controle ProcAmp, operações de desinterlace, operações de correção de proporção de aspecto, operações de conversão de espaço de cor, espelhamento vertical / horizontal e misturas alfa, o-  
10 perações de conversão de taxa de quadro e assim por diante, o acionador de dispositivo gráfico411 pode selecionar o provimento de toda ou de uma parte da amplitude operacional de processamento de vídeo, remanescente. Por meio da alocação de menos do que a totalidade da amplitude de operação de  
15 processamento de vídeo remanescente, o acionador de dispositivo gráfico422 é capaz de manter, em reserva, amplitudes de operação de processamento de vídeo adicionais para requisições subsequentes.

Uma troca de comunicação 504 se direciona ao es-  
20 tabelecimento de capacidades de propriedade de controle para uma operação de processamento de vídeo específica. Em uma requisição 504A, que é enviada a partir do renderizador de vídeo 410 ao acionador de dispositivo gráfico422, o renderizador de vídeo 410 especifica uma operação de processamento  
25 de vídeo em particular, alocada na resposta 502B. A requisição 504 também pode incluir uma consulta de que ou quais capacidades de propriedades o acionador de dispositivo gráfico422 é capaz de desempenhar em relação à operação, em par-

particular, de processamento de vídeo. Na resposta 504B, acionador de dispositivo gráfico422 informa ao renderizador de vídeo 410, em relação às capacidades de propriedade que se encontram disponíveis para a operação, em particular, especificada, de processamento de vídeo. A troca de comunicações 504 pode ser omitida, caso, por exemplo, não existam capacidades de propriedade de controle múltiplas para a operação de processamento de vídeo, em particular.

Uma troca de comunicação 504 refere-se ao estabelecimento de qual dentre as outras operações de processamento de vídeo alocadas pode ser executada simultaneamente com a operação de processamento de vídeo, em particular, conforme especificado. Em uma requisição 506A, o renderizador de vídeo 410 lança uma consulta ao acionador de dispositivo gráfico422 para determinar qual dentre as operações de processamento de vídeo, caso haja alguma, pode ser executada simultaneamente com a operação de processamento de vídeo, em particular. O acionador de dispositivo gráfico422 informa o renderizador de vídeo 410, na resposta 506B das operações de processamento de vídeo, que é possível ao acionador de dispositivo gráfico422 ser executado simultaneamente com a operação, em particular, de processamento de vídeo. À guisa de exemplo, deve ser observado que (i) transmissões 504A e 506A e/ou (ii) as transmissões 504B e 505B podem ser combinadas em uma única consulta e transmissão de resposta, respectivamente.

Uma troca de comunicação 508 refere-se ao estabelecimento de valores para uma propriedade especificada de

controle da operação de processamento de vídeo, em particular. Em uma requisição 508A, o renderizador de vídeo 410 especifica em uma consulta, uma propriedade de controle para a operação de processamento de vídeo, em particular. A propriedade de controle especificada pode ser selecionada dentre as propriedades de controle disponíveis, providas na resposta 504B. O acionador de dispositivo gráfico422 provê valores que se referem à propriedade de controle especificada para a operação de processamento de vídeo, em particular, com relação ao renderizador de vídeo 410. Estes valores podem ser pontos de ajuste numéricos, faixas, etc., que o renderizador de vídeo 410 pode utilizar como estrutura, quando instrui o acionador de dispositivo gráfico422 para execução da operação, em particular do processamento de vídeo. A troca de comunicação 508 pode ser repetida para cada propriedade de controle disponível, que se encontra indicada na resposta 504B. alternativamente, uma dentre estas trocas de comunicação 508 pode se referir a múltiplas propriedades de controle (incluindo todas estas) dentre as propriedades de controle disponíveis.

Uma troca de comunicação 510 refere-se ao início de um objeto de fluxo de processamento de vídeo. Em uma instrução 510A, um renderizador de vídeo 410 envia um comando a um acionador de dispositivo gráfico422 para abrir um objeto de fluxo de processamento de vídeo. Este comando pode ser transmitido em nome de uma aplicação ou de outro componente de software que esteja tentando apresentar imagens de vídeo no dispositivo de exibição 436. Na resposta 510B, o aciona-

dor de dispositivo gráfico422 retorna um dispositivo de manipulação para o objeto de fluxo de processamento de vídeo a um renderizador de requisição de vídeo 410.

Em uma transmissão 512 A, o renderizador de vídeo  
5 410 instrui o acionador de dispositivo gráfico422 a efetuar a operação, em particular, de processamento de vídeo alocada, ou outra operação alocada. O comando de realização de operação de processamento de vídeo pode incluir numerais selecionados para ajuste e/ou substituição de valores para uma  
10 ou mais propriedades de controle para a operação de processamento de vídeo, em particular. Em resposta, o acionador de dispositivo gráfico422 executa uma operação de processamento de vídeo 512B, como requisitado na transmissão 512A. tipicamente, pelo menos um renderizador de vídeo 410 se designa a  
15 cada aplicação, que deve ser exibida no vídeo. Sempre que uma aplicação instigante requisita uma operação de processamento de vídeo, por exemplo, um renderizador de vídeo 410, adianta tal requisição como uma instrução de operação de processamento de vídeo, opcionalmente após re - formatação,  
20 translação e assim por diante, ao acionador de dispositivo gráfico422.

A realização de comandos de operação de processamento de vídeo 512A e as operações de processamento de vídeo 512B resultantes, podem ser repetidas, conforme desejado,  
25 enquanto o objeto de fluxo de processamento de vídeo exista. Quando o vídeo se completa ou o software relevante termina, uma instrução de fechamento de objeto de fluxo de processamento

mento de vídeo 514 é transmitida a partir do renderizador de vídeo 410 ao acionador de dispositivo gráfico422.

As soluções das Figuras 4, 5 e 6, por exemplo, encontram ilustradas em diagramas que se dividem em múltiplos blocos e/ou múltiplas transmissões. No entanto, a ordem e/ou disposição com a qual as soluções estão descritas e/ou ilustradas não se designa a ser considerada como limitação e qualquer quantidade de blocos / transmissões pode se combinada e/ou re - arranjada em qualquer ordem para implementação de um ou mais sistemas, métodos, meios, protocolos, disposições etc., para facilitação da interação entre renderizadores de vídeo e acionadores de dispositivos gráficos. Além disso, embora a descrição neste caso inclua referências a implementações específicas, tais como as da Figura 4 (bem como o exemplo do ambiente de sistema da Figura 7) e aos exemplos APIs, as soluções podem ser implementadas em qualquer hardware, software firmware apropriados, ou misturas dos mesmos e com a utilização de qualquer/(quaisquer) linguagem/(ns) de programação, meio(s) de codificação, paradigma(s) de protocolo, ajuste(s) gráfico(s) apropriados e assim por diante.

#### Exemplo de Implementação Geral de API

A Figura 6 é um diagrama de fluxo 600, que ilustra um exemplo de método para facilitação entre um renderizador de vídeo 410 e um acionador de dispositivo gráfico422. Embora uma implementação descrita, como refletida pela Figura 6, se direcione a uma operação de ajuste ProcAmp, esta não se limita a tal. Ao contrário, pelo menos determinados

aspectos deste exemplo de implementação geral de APIs podem ser utilizados com uma ou mais operações de processamento de vídeo (ou imagem geral).

Em um diagrama de fluxo 600, o renderizador de vídeo 410 se associa a nove (9) blocos 602 - 618 e o acionador de dispositivo gráfico 422 fica associado a seis (6) blocos 620 - 630. Cada um dentre os blocos 602 - 618 e 620 - 630 corresponde a pelo menos uma ação que é efetuada por ou em nome do renderizador de vídeo 410 e do acionador de dispositivo gráfico 422, respectivamente.

O diagrama de fluxo 600 encontra-se descrito abaixo no contexto de um exemplo geral e APIs. Estas APIs, de forma geral, conforme descritas neste caso, podem ser divididas em dois grupos funcionais de métodos, lógica de dispositivos, etc. O primeiro grupo pode ser utilizado para determinar as capacidades de processamento de vídeo de um dispositivo gráfico. O segundo grupo pode ser utilizado para criar e fazer uso de objetos de fluxo de operação de processamento de vídeo.

Estes exemplos de APIs gerais, podem corresponder a APIs 416 (da Figura 4) que estão ilustrados como fazendo parte da interface de acionador de dispositivo 414, a qual suporta a interface gráfica 412 e faz interface com o acionador de dispositivo gráfico 422. As APIs 416, desta forma, se encontram ilustradas como fazendo parte de uma interface de acionador de dispositivo 414, que é uma parte de modo de usuário 418. No entanto, tais APIs 416 podem, alternativamente, ficar localizadas e/ou funcionando com outras lógicas

além da interface de acionador de dispositivo 414. Tal outra lógica inclui, à guisa de exemplo apenas, renderizador de vídeo 410, interface gráfica 412, alguma parte de parte em modo central 420 e assim por diante.

5           As APIs gerais, descritas abaixo nesta seção, podem ser utilizadas para estender / aperfeiçoar / etc., a aceleração de vídeo (VA) do Microsoft<sup>®</sup> DirectX<sup>®</sup>, por exemplo, de forma a apoiar quaisquer dentre uma quantidade de operações de processamento de vídeo (por exemplo, ajustes  
10 ProcAmp, conversões de taxa de quadro, etc.) para o conteúdo de vídeo sendo exibido em conjunto com um acionador de dispositivo gráfico. Informação adicional relacionada pode ser encontrada em uma Nota de Projeto de Plataforma Microsoft<sup>®</sup> Windows<sup>®</sup>, intitulada "DirectX<sup>®</sup> VA; Aceleração de Vídeo  
15 API/DDI"- "DirectX<sup>®</sup> VA; Video Acceleration API/DDI", datada de 23 de janeiro de 2001. "DirectX<sup>®</sup> VA; Aceleração de Vídeo API/DDI" encontra-se incorporada e este caso, em sua totalidade, à guisa de referência.

Embora as ações do diagrama de fluxo 600 encontrem-se descritas neste caso, em termos de APIs que podem ser aplicadas especificamente à evolução corrente dos sistemas de operação Microsoft<sup>®</sup> Windows<sup>®</sup>, para micro computadores, deve ser entendido que os blocos do mesmo, bem como outras implementações descritas neste caso, também podem ser  
25 aplicadas a outros sistemas de operação e/ou outros dispositivos eletrônicos.

Nos exemplos a seguir, a alimentação de saída da(a) operação(ões) de vídeo é provida em formato de rende-

rização de alvo RGB, tal como uma superfície alvo DirectDraw<sup>®</sup>. Em se fazendo isso, se exclui a necessidade de técnicas de sobreposição de hardware convencionais. Adicionalmente, ocorre a totalidade da tela, como visualizada em um dispositivo de exibição, incluindo quaisquer imagens de vídeo e, além disso, ficando presente em um local de memória, de forma que esta possa ser capturada por um comando de impressão. Esta captura de impressão de tela pode ser então passada a um documento, adicionada a um arquivo, diretamente impressa e assim por diante.

No diagrama de fluxo 600, o renderizador de vídeo 410 pode já ter sido informado pelo acionador de dispositivo gráfico 422 que o hardware gráfico associado é capaz de desempenhar operações de processamento de vídeo de ajuste ProcAmp ou o renderizador de vídeo 410 pode determinar a existência das capacidades ProcAmp, ou a falta destas, como se segue. No bloco 620, renderizador de vídeo 410 provê uma descrição do vídeo a ser exibido e requisita capacidades de processamento gráfico em relação às propriedades de controle ProcAmp.

O renderizador de vídeo 410 executa a provisão de descrição de vídeo e/ou a requisição das propriedades de controle ao acionador de dispositivo gráfico 422, através de uma ou mais transmissões, como indicado pela seta de transmissão, entre o bloco 602 e o bloco 620. A descrição do vídeo possibilita ao acionador de dispositivo gráfico 422 desenvolver as capacidades disponíveis / possíveis / etc., de processamento de vídeo com base no tipo de vídeo. Por exem-

plo, um conjunto predeterminado de capacidades pode ser constituído para cada um dentre os diferentes tipos de vídeo.

No bloco 620, o acionador de dispositivo gráfico422 provê ao renderizador de vídeo 410 uma listagem das propriedades de controle ProcAmp disponíveis. Esta lista pode incluir nenhum, um ou mais de um dentre brilho, contraste, matiz e saturação. No bloco 604, o renderizador de vídeo 410 recebe as propriedades de controle ProcAmp disponíveis a partir do acionamento de dispositivo gráfico422. As ações dos blocos 620 e 622 podem ser efetuadas em resposta à(s) comunicação(ões) do bloco 602. Alternativamente, o renderizador de vídeo 410 pode efetuar uma consulta em separado para esclarecer as ações do bloco 622.

No bloco 622, o acionador de dispositivo gráfico422 provê o renderizador de vídeo 410 com aquelas operações de processamento de vídeo que podem, possivelmente ser realizadas simultaneamente / concomitantemente com as operações de ajuste ProcAmp. Tais operações de processamento de vídeo podem incluir nenhuma, uma ou mais dentre YUV2RGB, S-tretchX, StretchY, SubRects e AlphaBlend. Outras dentre tais operações de processamento de vídeo podem incluir desinterlace, conversão de taxa de quadro e assim por diante. No bloco 606, o renderizador de vídeo 410 recebe as operações de processamento de vídeo simultâneas possíveis a partir do acionador de dispositivo gráfico422.

Um exemplo de API geral, para implementação de pelo menos parte das ações dos blocos 602, 606, 620 e 622 é provido como se segue:

```
ProcAmpControlQueryCaps
```

5 Esta API possibilita ao renderizador de vídeo 410 indagar o acionador de dispositivo gráfico<sup>422</sup> para determinar a informação relacionada às exigências de alimentação de entrada de um dispositivo de controle ProcAmp e quaisquer operações de processamento de vídeo adicionais que possam ser suportadas ao mesmo tempo em que as operações de ajuste ProcAmp estão sendo realizadas.

```
HRESULT
```

```
ProcAmpControlQueryCaps(  
[in] DXVA_VideoDesc*lpVideoDescription  
15 [out] DXVA_ProcAmpControlCaps*lpProcAmpCaps  
);
```

O acionador de dispositivo gráfico<sup>422</sup> anuncia suas capacidades para aquele modo em uma alimentação de saída de estrutura DXVA\_ProcAmpControlCaps para lpProcAmpCaps.

```
20 typedef struct DXVA_ProcAmpControlCaps  
{DWORD Size;  
DWORD InputPool;  
D3DFORMAT OutputFrameFormat;  
DWORD ProcAmpControlProps;  
25 DWORD VideoProcessingCaps;}  
DXVA_ProcAmpControlCaps;
```

O campo Size (dimensão) indicando a dimensão da estrutura de dados e podendo ser utilizado, entre outros,

como um indicador de versão, caso versões diferentes apresentem diferentes dimensões de estrutura de dados.

O campo `InputPool` indica uma mistura de memória, a partir da qual as superfícies de fonte de vídeo devem ser alocadas. Por exemplo, a mistura de memória pode estar localizada em uma memória de vídeo local na placa gráfica, em uma memória de sistema especialmente etiquetado (por exemplo, memória de portal gráfico acelerado (AGP)), memória de sistema geral e assim por diante. As documentações D3D e `DirectDraw` também provêm uma descrição de locações de misturas de memória válidas.

O campo `OutputFrameFormat` indica um formato de superfície `Direct3D` das estruturas de alimentação de saída. O dispositivo `ProcAmp` pode fornecer estruturas de alimentação de saída em um formato de superfície que se conjuga ao formato de superfície de alimentação de entrada. Este campo assegura que o renderizador de vídeo 410 será capaz de suprir o formato correto para as superfícies de estrutura de alimentação de saída e, relação ao hardware de controle `ProcAmp`. Observe-se que caso a bandeira `DXVA_VideoProcess_YUV2RGB` (vide abaixo) seja retornada no campo `VideoProcessingCaps`, o renderizador de vídeo 410 pode assumir que os formatos de alimentações de saída válidos sejam especificados por este campo, bem como por um formato RGB, tal como `RGB32`. O `RGB32` trata-se de um formato RGB com 8 bits de precisão para cada dentre os canais Vermelho, Verde e Azul e 8 bits de dados não utilizados.

O campo ProcAmpControlProp identifica operações ProcAmp que o hardware é capaz de executar. O acionador de dispositivo gráfico<sup>422</sup> retorna a lógica da mistura de operações ProcAmp que este apoia:

5 . DXVA\_ProcAmp\_None. O hardware não apoia nenhuma operação de controle ProcAmp.

. DXVA\_ProcAmp\_Brightness. O hardware de controle ProcAmp desempenha ajustes de brilho na imagem de vídeo.

10 . DXVA\_ProcAmp\_Contrast. O hardware de controle ProcAmp desempenha ajustes de contraste na imagem de vídeo.

. DXVA\_ProcAmp\_Hue. O hardware de controle ProcAmp desempenha ajustes de matiz na imagem de vídeo.

. DXVA\_ProcAmp\_Saturation. O hardware de controle ProcAmp desempenha ajustes de saturação na imagem de vídeo.

15 O campo VideoProcessingCaps identifica outras operações que podem ser realizadas de maneira concorrente ao ajuste ProcAmp requisitado. As seguintes bandeiras identificam as operações possíveis:

20 . DXVA\_VideoProcess\_YUV2RGB. O hardware de controle ProcAmp pode converter o vídeo a partir do espaço de cor YUV para um espaço de cor RGB. O formato RGB utilizado pode possuir 8 bits ou mais de precisão para cada componente de cor. Caso isso seja possível, uma cópia de amortecimento no renderizador de vídeo pode ser evitada. Observe-se que  
25 não existem exigências em relação a esta bandeira para conversão de um espaço de cor RGB para um espaço de cor YUV.

. DXVA\_VideoProcess\_StretchX. Caso o hardware de controle ProcAmp seja capaz de se expandir ou contrair de

forma horizontal, uma correção de proporção de aspecto pode ser realizada ao mesmo tempo em que o vídeo estiver sendo ajustado de forma ProcAmp.

. DXVA\_VideoProcess\_StrechY. Algumas vezes o ajuste de proporção de aspecto combina-se à operação de redimensionamento de quadro geral, para graduar a imagem de vídeo dentro um espaço de composição definido por aplicação. Isso, algumas das vezes, trata-se de uma característica rara. A realização de graduação para redimensionamento de vídeo para adaptação à janela de aplicação pode ser efetuado ao mesmo tempo em que a graduação para o ajuste ProcAmp. A realização destas graduações, em conjunto, evita artefatos cumulativos.

DXVA\_VideoProcess\_SubRects. Esta bandeira indica que o hardware é capaz de operar em uma (sub) região retangular da imagem, bem como na totalidade da imagem. A região retangular pode ser identificada por um retângulo gráfico de fonte em uma estrutura de dados DXVA\_ProcAmpControlBlt.

DXVA\_VideoProcess\_AlphaBlend. A mistura alfa pode controlar como outras informações gráficas são exibidas, tais como por meio de ajuste de níveis de transparência ou de opacidade. Assim sendo, um valor alfa pode indicar a transparência de uma cor - ou a extensão com a qual a cor se combina com qualquer cor de fundo. Tais valores alfa podem variar de uma cor completamente transparente a uma cor totalmente opaca.

Em operação, a mistura alfa pode ser concretizada com a utilização de uma mistura pixel - por - pixel de dados de cor de fonte e fundo. Cada uma dentre os três componentes de cores (vermelho, verde e azul) de uma dada fonte de cor  
5 podem ser combinados com o componente correspondente de uma cor de fundo, para execução de uma operação de mistura alfa. Em um exemplo de implementação, a cor pode, de forma geral, ser representada por um valor de 32 bits com 8 bits para cada alfa, vermelho, verde e azul.

10 Novamente, com a utilização desta característica pode-se evitar uma cópia de amortecimento com o renderizador de vídeo 410. No entanto, esta também se trata de uma característica raramente utilizada, devido ao fato das aplicações, raramente alterarem o valor constante de alfa, associado ao seu feixe de vídeo.  
15

No bloco 608 do diagrama de fluxo 600, o renderizador de vídeo 410 seleciona uma propriedade de controle ProcAmp a partir das recebidas no bloco 604. No bloco 610, o renderizador de vídeo 410 requisita um ou mais valores para  
20 a propriedade de controle ProcAmp, selecionada, a partir do acionador de dispositivo gráfico422. No bloco 624, o acionador de dispositivo gráfico422 envia ao renderizador de vídeo 410, valores para a propriedade de controle ProcAmp requisitada. Tais valores referem-se a um ou mais dentre o valores  
25 padrões, valores de incremento, valores mínimos, valores máximos e assim por diante.

No bloco 612, o renderizador de vídeo 410 recebe, a partir do acionador de dispositivo gráfico422 e com isso,

é informado de um ou mais valores para a propriedade de controle ProcAmp selecionada. Como indicado pela seta de fluxo, a partir do bloco 612 ao bloco 608, as ações dos blocos 608, 610, 612 e 624 podem ser repetidas por mais de uma, incluindo todas as propriedades de controle ProcAmp disponíveis. Alternativamente, o renderizador de vídeo 410 pode indagar o acionador de dispositivo gráfico422 por mais de uma vez, incluindo todas as propriedades de controle ProcAmp disponíveis em uma única troca de comunicação, apresentando uma ou mais transmissões.

Um exemplo de API geral, para implementação de pelo menos parte das ações dos blocos 608, 610, 612 e 624 se provê, conforme se segue:

ProcAmpControlQueryRange

Para cada propriedade ProcAmp (brilho, contraste, saturação, matiz, etc.) o renderizador de vídeo 410 indaga o acionador de dispositivo gráfico422, para determinação de valor mínimo, máximo, dimensão de etapa (incremento), padrão e assim por diante. Caso o hardware não apoie uma propriedade de controle ProcAmp, em particular, o acionador de dispositivo gráfico422 pode retornar "E\_NOTIMPL", em resposta à função ProcAmpControlQueryRange.

Embora o acionador de dispositivo gráfico422 possa retornar quaisquer valores que deseje para as diferentes propriedades de controle ProcAmp, os seguintes valores de ajuste são providos à guisa de exemplo (todos os valores tabulados sendo flutuantes):

<u>Propriedade</u>	<u>Mínimo</u>	<u>Máximo</u>	<u>Padrão</u>	<u>Incremento</u>
--------------------	---------------	---------------	---------------	-------------------

Brilho	- 100,0F	100,0F	0,0F	0,1F
Contraste	0,0F	10,0F	1,0F	0,01F
Saturação	0,0F	10,0F	1,0F	0,01F
Matiz	-180,0F	180,0F	0,0F	0,1F

Caso os valores padrões resultem em uma transformação zero do feixe de vídeo, se permite que o renderizador de vídeo 422 desvie do estágio de ajuste ProcAmp em sua encadeamento de vídeo, caso a aplicação instigante não altere  
5 quaisquer dentre as propriedades de controle ProcAmp.

HRESULT

```
ProcAmpControlQueryRange(
    [in] DWORD_VideoProperty,
    [in] DXVA_VideoDesc*lpVideoDescription
10    [out] DXVA_VideoPropretyRange*lpPropRange
    );
```

VideoProperty identifica a propriedade (ou as propriedades) de controle ProcAmp que forma requisitadas ao acionador de dispositivo de controle 422 para retorno de in-  
15 formação. Em uma implementação descrita, ao valores dos parâmetros possíveis para este campo são:

- . DXVA\_ProcAmp\_Brightness (brilho)
- . DXVA\_ProcAmp\_Contrast (contraste)
- . DXVA\_ProcAmp\_Hue (matiz)
20 - . DXVA\_ProcAmp\_Saturation (saturação)

lpVideoDesdription provê o acionador de dispositivo gráfico 422 com uma descrição do vídeo ao qual o ajuste ProcAmp irá ser aplicado. O acionador de dispositivo gráfi-

co422 pode ajustar sua característica de apoio ProcAmp para tipos de descrição de feixes de vídeo específicos.

LpPropRange identifica os valores de alcance (mínimo e máximo), dimensão de etapa e padrão para a propriedade de controle ProcAmp que foi especificada pelo campo / parâmetro VideoProperty.

```

    typedef struct_DXVA_VideoPropertyRange
    {
        FLOAT MinValue;
        FLOAT MaxValue;
10      FLOAT DefaultValue;
        FLOAT StepSize;}
        DXVA_VideoPropertyRange*LPDXVA_VideopropertyRange;

```

No bloco 614 do diagrama 600, o renderizador de vídeo 410 envia um comando de objeto de fluxo ProcAmp aberto ao acionador de dispositivo gráfico422. Em resposta, o acionador de dispositivo gráfico422 abre um objeto de fluxo ProcAmp no bloco 626. No bloco 616, o renderizador de vídeo 410 instrui o acionador de dispositivo gráfico422 a desempenhar uma operação de ajuste ProcAmp. Em resposta, o acionamento de dispositivo gráfico422 desempenha a operação de ajuste ProcAmp requisitada no bloco 628.

Como indicado pela seta de fluxo curva, no bloco 616, o renderizador de vídeo 410 pode continuar a enviar instruções de operação de ajuste de realização ProcAmp ao acionador de dispositivo gráfico422, por tanto tempo quanto desejado (por exemplo, sempre que exigido por uma aplicação de instigação exibido o feixe de vídeo). No bloco 618, o

renderizador de vídeo 410 instrui o acionador de dispositivo gráfico422 a fechar o objeto de fluxo ProcAmp. O acionador de dispositivo gráfico422 fecha então o objeto de fluxo ProcAmp no bloco 630.

5 Um exemplo de API geral para implementação de pelo menos parte das ações dos blocos 614, 618, 626, 628 e 630 é provido conforme se segue:

The ProcAmpStreamObject

Após o renderizador de vídeo 410 ter determinado  
10 as capacidades do hardware de controle ProcAmp, um objeto ProcAmpStream pode ser criado. A concretização de um objeto ProcAmpStream permite que o acionador de dispositivo gráfico422 reserve quaisquer recursos que sejam necessários para realização da(s) operação(ões) de ajuste ProcAmp requisita-  
15 das.

ProcAmpOpenStream

O método ProcAmpOpenStream cria o objeto ProcAmpStream.

HRESULT

20 ProcAmpOpenStream(  
[in] LPDXVA\_VideoDesc\*lpVideoDescription,  
[out] HDXVA\_ProcAmpStream\*lpHCCStrm  
);

O parâmetro de alimentação de saída  
25 HDXVA\_ProcAmpStream é um dispositivo de manipulação para o objeto ProcAmpStream, sendo utilizado neste feixe em chamadas futuras que se direcionam ao mesmo.

## ProcAmpBlt

O método ProcAmpBlt desempenha a operação de ajuste ProcAmp escrevendo a alimentação de saída para a superfície de destino, durante uma operação de transferência  
5 de bloco de bits.

HRESULT

```
ProcAmpBlt(
    [in] HDXVA_ProcAmpStreamhCcStrm
    [in] LPDDSURFACElpDDSDstSurface,
10    [in] LPDDSURFACElpDDSSrcSurface,
    [in] DXVA_ProcAmpBlt*ccBlt
);
```

Os retângulos de fonte e de destino são utilizados para um ajuste ou para uma extensão de sub - retângulo  
15 de ProcAmp. O suporte para a extensão é opcional (sendo anunciado pelas bandeiras Caps). Do mesmo modo, o suporte para o sub - retângulo não é mandatário.

A superfície de destino pode ser um plano fora de tela, um alvo de renderização D3D, uma textura de acordo com  
20 a reivindicação D3D, que também se trata de um alvo de renderização, e assim por diante. A superfície de destino pode ser alocada em uma memória de vídeo local, por exemplo. O formato de pixel da superfície de destino é um dentre os indicados na estrutura DXVA\_ProcAmpCaps, a não ser que uma  
25 conversão de espaço de cor YUV para RGB esteja sendo realizada, em conjunto com uma operação de ajuste ProcAmp. Nestes

casos, o formato da superfície de destino é um formato RGB com 8 bits ou mais de precisão, para cada componente de cor.

ProcAmpCloseStream

O método ProcAmpClose Stream fecha o objeto ProcAmpStream e instrui um acionador de dispositivo gráfico a liberar quaisquer recursos de hardware, associados com o feixe identificado.

HRESULT

ProcAmpCloseStream(

10 HDXVA\_ProcAmpStreamhCcStrm

);

#### Exemplo de Implementação API Específica

A situação particular e os exemplos de APIs descritos abaixo nesta seção, podem ser especificamente aplicados a um subconjunto de sistemas de operação existentes Microsoft® Windows®, para microcomputadores. No entanto, deve, contudo, ser entendido que os princípios, bem como determinados aspectos do pseudo código, que se encontram presentes abaixo, podem ser utilizados (propriamente ditos ou com modificações de rotina), em conjunto com outros sistemas de operação e/ou outros ambientes.

#### Mapeamento DDI para uma Interface ProcAmp

Para compatibilidade com a infra-estrutura DDI, para um subconjunto de sistemas de operação existentes Microsoft® Windows®, a API descrita acima, na seção anterior, pode ser "mapeada" em relação aos DDI para DirectDraw e DirectX VA existentes. Esta seção descreve um mapeamento de

interface ProcAmp em relação aos DirectDraw e DX - VA DDI existentes.

O DX - VA DDI, propriamente dito, trata-se de dois grupos funcionais: o "recipiente DX - VA" e o "dispositivo DX - VA". A finalidade do recipiente DX - VA do grupo DDI é a de determinação da quantidade e das capacidades dos vários dispositivos DX - VA, contidos no hardware de exibição. Assim sendo, um acionamento DX - VA pode apenas apresentar um único recipiente, mas pode apoiar múltiplos dispositivos DX - VA.

Não é factível um mapeamento da chamada de ProcAmpCaps em quaisquer um dentre os pontos de entrada DDI no grupo de recipientes DX \_ VA, devido, ao contrário do restante de DX - VA, os métodos de recipientes utilizam parâmetros digitados. No entanto, o dispositivo DX - VA do grupo DDI não utiliza parâmetros digitados, assim sendo factível o mapeamento da interface de controle ProcAmp em relação aos métodos no grupo de dispositivos. Esta seção descreve um exemplo específico de como uma interface ProcAmp pode ser mapeada em relação ao dispositivo DX - VA de DDI.

#### Dispositivo de Desinterlace de Recipiente

Os métodos dos dispositivos DX - VA não utilizam parâmetros digitados, desta forma, estes métodos podem ser reutilizados para diversas finalidades diferentes. No entanto, os métodos dos dispositivos DX - VA podem ser apenas utilizados no contexto do dispositivo DX - VA, de maneira a definir e criar um "dispositivo de recipiente" especial.

A Aplicação de Patente Norte Americana, não Provisória, Nº de série 10/273.505, que se intitula "Métodos e Dispositivos para Facilitação de Processamento de Imagens de Vídeo Interlaçadas para Exibições de Vídeo Progressivo" -  
5 "Methods and Apparatuses for Facilitating Processing of Interlaced Video Images for Progressive Video Displays" e que se encontra incorporada a este caso à guisa de referência, inclui uma descrição do dispositivo de desinterlace de recipiente. Este dispositivo de desinterlace de recipiente, des-  
10 crito na Aplicação mencionada é reutilizado para a função ProcAmpQueryCaps.

O dispositivo de desinterlace DX - VA é um software, construído apenas de forma que não represente qualquer hardware contido no dispositivo físico. O pseudo código do  
15 acionador (dispositivo) de amostragem de controle ProcAmp apresentado abaixo, indica como o dispositivo de recipiente pode ser implementado por um acionador.

Chamada de DDI, Proveniente de um Componente de Modo de Usuário

20 Um exemplo de seqüência de oito (8) tarefas que utilizam DDI a partir de um componente de modo de usuário, tal como um renderizador (de vídeo), apresenta-se conforme de segue:

1. Chamada GetMoCompGuids para alcance da lista-  
25 gem de dispositivos DX - VA, apoiados pelo acionador.

2. Caso o "dispositivo de desinterlace de recipiente" GUID encontre-se presente, chamada de CreateMoComp pa-

ra geração de um exemplo deste dispositivo DX - VA. O dispositivo de recipiente GUID se define conforme se segue:

```

DEFINE_GUID(DXVA_DeinterlaceContainerDevice,
            0x0e85cb93, 0x3046, 0x4ff0, 0xae, 0xcc, 0xd5, 0x8c, 0xb5,
5 0xf0, 0x35, 0xfc);

```

3. Chamada de RenderMocomp com uma parâmetro dwFunction que identifica uma operação ProcAmpControlQueryModeCaps. Novamente, o parâmetro lpInputdata é utilizado para passagem dos parâmetros de alimentação de entrada ao acionador, o qual retorna sua alimentação de saída através do parâmetro lpOutputData.

4. para cada propriedade de ajuste ProcAmp, apoiada por hardware, o renderizador chama RenderMocomp em relação ao parâmetro dwFunction, que identifica uma operação ProcAmpControlQueryRange. O parâmetro lpInputData é utilizado para passagem dos parâmetros de alimentação de entrada ao acionador, o qual retorna sua alimentação de saída através do parâmetro lpOutputData.

5. Após o renderizador ter determinado as capacidades de ajuste parte do hardware, este chama CreateMocomp para criar um exemplo de dispositivo de controle ProcAmp. O dispositivo de controle ProcAmp GUID é definindo como se segue:

```

DEFINE_GUID(DXVA_ProcAmpControlDevice,
25 0x9f200913, 0x2ffd, 0x4056, 0x9f, 0x1e, 0xe1, 0x5, 0x08, 0
xf2, 0x2d, 0xcf);

```

6. O renderizador chama então RenderMocomp do dispositivo de controle ProcAmp com um parâmetro de função

DXVA\_ProcAmpControlBlitFnCode para cada operação de ajuste ProcAmp.

7. Quando o renderizador não mais necessita desempenhar mais operações ProcAmp, este chama DestroyMocomp.

5 8. O acionador libera quaisquer recursos utilizados pelo dispositivo de controle ProcAmp.

#### ProcAmpControl QueryCaps

Este método mapeia diretamente em relação à chamada de método RenderMoComp do dispositivo de desinterlace de recipiente. A estrutura DD\_RENDERMOCOMPDATA se completa, conforme se segue:

- . dwNumBuffer é zero
- . lpBufferInfo é NULA
- . dwFunction se define como

15 DXVA\_ProcAmpControlQueryCapsFnCode

- . lpData aponta a uma estrutura DXVA\_VideoDesc
- . lpOutputData aponta a uma estrutura DXVA\_ProcAmpCaps.

Observe-se que o método RenderMoComp do dispositivo de recipiente DX - VA pode ser chamado sem BeginMoCompFrame ou EndMoCompFrame serem inicialmente chamados

#### ProcAmpControlQueryRange

Este método mapeia diretamente em relação a uma chamada ao método RenderMoComp do dispositivo de interlace de recipiente. A estrutura DD\_RENDERMOCOMPDATA se completa conforme se segue;

- . dwNumBuffer é zero
- . lpBufferInfo é NULA

. dwFunction se define como  
 DXVA\_ProcAmpControlQueryRangeFnCode  
 . lpInputData aponta a uma estrutura  
 DXVA\_ProcAmpControlQueryRange.

```

5      Typedef struct_DXVA_ProcAmpQueryRange {
        DWORDgráficoSize; (dimensão)
        DWORDgráficoVideoProperty;
        DXVA_VideoDesc VideoDesc;
        } DXVA_ProcAmpControlQueryRange,
10     *LPDXVA_ProcAmpControlQueryRange;

```

. lpOutputData irá apontar a uma estrutura  
 DXVA\_VideoPropertyRange.

Observe-se que o método RenderMoComp de dispositivo de recipiente DX - VA pode ser chamado sem BeginMoComp-  
 15 Frame ou EndMoCompFrame serem inicialmente chamados.

ProcAmpControOpenStream

Este método mapeia diretamente em relação ao método CreateMoComp da estrutura DD\_MOTIONCOMPCALLBACKS, em que o GUID se trata do GUID ProcAmp Device (dispositivo), pUn-  
 20 compData apontando a uma estrutura que na contenha nenhum dado (todos zero) e pData apontando à estrutura DXVA\_VideoDesc.

Caso um acionador apoie decodificação acelerada de vídeo comprimido, o renderizador pode chamar o acionador para  
 25 geração de dois dispositivos DX - VA --- uma para realização do serviço real de decodificação, como definido pela

especificação DirectX VA Video Decoding e o outro para utilizado estritamente para ajuste ProcAmp.

**\*\* Exemplo: Mapeamento de CreateMoComp em relação a ProcAmpControlOpenStream \*\***

5           O exemplo pseudo código abaixo ilustra como um acionador pode mapear a chamada CreateMoComp DDI em chamadas para ProcAmpControlOpenStream. O pseudo código ilustra como a função CreateMocComp é utilizada para ProcAmp. Caso o acionamento apoie outras funções DX - VA, tal como decodificação de feixes de vídeo MPEG-2, a amostra de código abaixo  
10       pode ser estendida para inclusão de processamento de DX-VA GUIDs adicionais.

```

        DWORD APIENTRY
        CreateMoComp(
15         LPDDHAL_CREATEMOCOMPDADAlpData
        )
        {
            // Make sure its a guid we like.
            (assegure-se que se trata de um guid de que gos-
20         tamos)

            caso (!ValidDXVAGuid(lpData->lpGuid)) {

                DbgLog((LOG-ERROR, 1,
                TEXT ("No formats supported for this GUID")));
                (texto: nenhum formato apoiado por este GUID)

25         lpData->ddRVal = E_INVALIDARG;
            retornar DDHAL_DRIVER_HANDLED;
    
```

```

    }

    //Look for the deinterlace container device GUID
    (procure o dispositivo de desinterlace de recipi-
ente GUID)
5      caso(*lpData->lpGuid == DXVA_Deinterlace Contai-
nerDevice

        DXVA_DeinterlaceContainerDeviceClass**lpDev =
        Novo DXVA_DeinterlaceContainerdeviceClass(
        *lpData->lpGuid,
10     DXVA_DeviceContainer);

        caso(lpDev) {
        lpData->ddRVal = DD_OK;
        {
        no mais {
15     lpData->ddRVal = E_OUTOFMEMORY;
        }
        lpData->.lpMoComp->lpDriverReserved1=
        (LPVOID) (DXVA_DeviceBaseClass*) lpDev;
        retorno DDHAL_DRIVER_HANDLED;
20     }

        // Look for the ProcAmp Control device GUID
        (procure o dispositivo de controle GUID do Pro-
cAmp)

        caso(*lpData->lpGuid== DXVA_ProcAmpControlDevice)
25     {

```

```

DXVA_ProcAmpControlDeviceClass*lpDev =
novo DXVA_ProcAmpControlDeviceClass(
*lpData->lpGuid,
DXVA_DeviceProcAmpControl);

5      Caso(lpDev) {
      LPDXVA_VideoDesc lpVideoDescription =
      (LPDXVA_VideoDesc)lpData->lpData;
      lpData->ddRVal=
      lpDev->ProcAmpControlOpenStream(
10     lpVideoDescription);

      caso (lpData->ddRval! = DD_OK) {
      deletar lpDev;
      lpdev = NULO
      }
15     }

      no mais {
      lpData->ddRVal = E_OUTOFMEMORY;
      }
      lpData->lpMoComp->lpDriveReserved 1 =
20     (LPVOID) (DXVA_DeviceBaseClass*)lpDev;
      retorne DDHAL_DRIVER_HANDLED;
      }

      lpData->ddRVal = DDERR_CURRENTLYNOTAVAIL;
      retorno DDHAL_DRIVER_HANDLED;
25     }

```

**\*\* Exemplo: Implementação de GetMoCompGuids\*\***

Em adição à criação da função CreateMoComp DDI, um acionador também pode ser capaz de implementar o método GetMoCompGuids da estrutura DD\_MOTIONCOMPCALLBACKS. O seguinte exemplo de pseudo código ilustra uma maneira de implementação desta função em um acionador.

```

//esta é uma listagem do dispositivo DD-VA, apoi-
ado por GUID
//o acionador - esta lista inclui decodificador
10 ProcAmp e
//o dispositivo de desinterlace. Não existe sig-
nificado para
//a ordem do GUID na listagem.
DWORD g_dwDXVANumSupportedGUIDs = 2
15 Const GUID*g_DXVASupportedGUID[2] = {
&DXVA_Deinterlacecontainerdevice
&DXVA_ProcAmpControlDevice
};
DWORDAPIENTRY
20 GetMoCompGuids(
PDD_GETMOCOMPGUIDSDATA lpData
)
{
DWORD dwNumToCopy;
25 //verifique para constatar se é uma requisição
GUID ou uma requisição de contagem
caso(lpData->lpGuids){

```

```

        dwNumToCopy =
        min(g_dwDXVANumSupportedGUIDs,
        lpData->dwNumGuids),
        for (DWORD i = 0; i <dwNumToCopy; i++){
5         lpData->lpGuids[i]=
            *g_DXVASupportedGUIDs[i]
        }
        }
        no mais }
10         dwNumToCopy = g_dwDXVANumSupportedGUIDs;
        }
        lpData->dwNumGuids = dwNumToCopy;
        lpData->ddRVal = DD_OK
        retorno DDHAL_DRIVER-HANDLED;
15     }

```

#### ProcAmpControlBlt

Este método mapeia diretamente a um método RenderMoComp da estrutura DD\_MOTIONCOMPCALLBECKS, em que:

```

    . dwNumBuffers é dois.
20     . lpBufferInfo aponta a um arranjo de duas superfícies. O primeiro elemento deste arranjo é a superfície de destino; o segundo elemento do arranjo trata-se da superfície de fonte.
    . dwFunction se define como
25     DXVA_ProcAmpControlBltCode.
    . lpInputData aponta para a seguinte estrutura:
    typedef struct_DXVA_ProcAmpControlBlt {

```

DWORD	Dimensão;
RECT	DstRect;
RECT	SrcRect;
FLOAT	Alfa
FLOAT	Brilho
FLOAT	Contraste
FLOAT	Matiz
FLOAT	Saturação

```
} DXVA_ProcAmpControlBlt;
```

.lpOutPutData é NULO

Observe-se que o dispositivo DX - VA utilizado para ProcAmp, RenderMoComp pode ser chamado sem chamada de BeginMoCompFrame ou de EndMoCompFrame.

**\*\* Exemplo: mapeamento RenderMoComp em relação a ProcAmpControlBlt\*\***

O exemplo de pseudo código utilizado abaixo, ilustra como um acionador pode mapear a chamada RenderMoComp DDI em chamadas ProcAmpBlt. O mesmo código de amostragem ilustra como a função RenderMoComp é utilizada para ajuste ProcAmp. Caso o acionador apoie outras funções DX - VA, tais como decodificação de feixes de vídeo MPEG-2, o código de amostragem abaixo pode ser estendido para inclusão de processamento de DX - VA GUIDs adicionais.

```
DWORD APIENTRY
```

```
RenderMoComp (
```

```
LPDDHAL_RENDERCOMPDATA lpData
```

```

    )
    {
        caso (lpData->dwFunction ==
DXVA_ProcAmpControlBltnFnCode)
5      {
        DXVA_ProcAmpControlDeviceClass*pDXVADev =
        (DXVA_ProcAmpControlDeviceClass*)pDXVABase;
        DXVA_ProcAmpControlBltn*lpBltn =
        (DXVA_ProcAmpControlBltn*)lpData->lpInputData;
10      LPDDMCBUFFERINFO lpBuffInfo = lpData-
        >lpBufferInfo;

        lpData->ddRVal = pDXVADev->ProcAmpControlBltn(
        lpBuffInfo[0].lpCompSurface,
        lpBuffInfo[1].lpCompSurface,
15      lpBltn);
        retorna DDHAL_DRIVER_HANDLED
    }
    lpData->ddRVal = E_INVALIDARG;
    retorno DDHAL_DRIVER_HANDLED;
20 }

```

ProcAmpControlCloseStream

Este método mapeia diretamente a um método DestroyMoComp da estrutura DD\_MOTIONCOMPCALLBACKS.

**\*\* Exemplo: Mapeamento DestroyMoComp a ProcAmp-**  
**ControlCloseStream\*\***

O seguinte exemplo de pseudo código ilustra como um acionador pode mapear a chamada destroyMoComp DDI em cha-

madas ProcAmpControlClassStream. O código de amostragem ilustra como a função DestroyMoComp é utilizada para controle ProcAmp. Caso o acionador apoie outras funções DX - VA, tais como decodificação de feixes de vídeo MPEG-2, o código de amostragem abaixo pode ser estendido para inclusão do processamento de DX-VA GUIDs adicionais.

```

    DWORDAPIENTRY
    DestroyMoComp(
    LPDDHAL_DESTROYMOCOMPDATA lpdata
10      )
      {
        DXVA_DeviceBaseClass*pDXVABase =
        (DXVA_DeviceBaseClass*)
        lpData->lpMoComp->Driverreserved1;

15      caso (pDXVABase == NULO) {
        lpData->ddRVal = E_POINTER
        retorno DDHAL_DRIVER_HANDLED;
        }

        comutar (pDXVABase->m_DeviceType) {
20      caso DXVA_DeviceContainer:
        lpData->ddRVal = S_OK;
        deletar pDXVABase;
        break;

        caso DXVA_Device_ProcAmpControl:
25      {
  
```

```

DXVA_ProcAmpControlDeviceClass*pDXVADev =
(DXVA_ProcAmpControlDeviceClass*)pDXVABase;
lpData->ddRVal = pDXVADev-
>ProcAmpControlStream();
5   deletar pDXVADev;
    }
    break
    } retorno DDHAL_DRIVER_HANDLED
    }

```

#### 10 Exemplo de Ambiente de Operação para Computador ou Outro Dispositivo Eletrônico

A Figura 7 ilustra um exemplo de ambiente de computação (ou dispositivo eletrônico em geral) 700, que é capaz se implementar (completa ou parcialmente), em pelo menos um sistema, dispositivo, componente, disposição, protocolo, solução, método, processo algumas misturas destes, etc., para 15 facilitação da interação entre renderizadores de vídeo e acionadores de dispositivos gráficos, como descritos neste caso. O ambiente de computação 700 pode ser utilizado em arquiteturas de computador e de ambiente de rede, descritas 20 abaixo e em situação de localização isolada.

O exemplo de um ambiente de operação de dispositivo eletrônico 70 é apenas um exemplo de ambiente e não se designa a sugerir quaisquer limitações ao escopo de uso ou 25 funcionamento das arquiteturas eletrônicas aplicáveis (incluindo computador, console de jogos, televisão, etc.). Tampouco o ambiente de dispositivo eletrônico 700 deveria ser

interpretado com apresentando quaisquer dependências ou exigências em relação a qualquer um dentre os componentes ou a qualquer mistura de componentes, conforme ilustrado na Figura 7.

5                   Adicionalmente, a facilitação da interação entre renderizadores de vídeo e acionadores de dispositivos gráficos podem ser implementada com numerosos outros ambientes ou configurações de dispositivos eletrônicos de finalidades gerais ou de finalidades especiais (incluindo sistemas de com-  
10   putação). Exemplos de sistemas, ambientes e/ou configurações (de dispositivos) eletrônicos que podem ser apropriados para uso, incluem, mas não se limitam a, microcomputadores, computadores de serviço, clientes esbeltos, clientes espessos, assistentes pessoais digitais (PDAs) ou telefones móveis,  
15   dispositivos seguros na mão ou do tipo laptop, sistemas de processamento múltiplo, sistemas com base em micro processamento, caixas de ajuste de topo, dispositivos eletrônicos de consumo programáveis, máquinas de vídeo game, consoles de jogos, unidades de jogo portáteis ou de manipulação, PCs de  
20   rede, mini computadores, computadores de estrutura principal, ambientes de computação distribuída que incluam qualquer dentre os sistemas ou dispositivos, misturas dos mesmos e assim por diante.

                  Implementações para facilitação de interação entre  
25   renderizadores ou representantes de vídeo e acionadores de dispositivos gráficos podem ser descritos, no contexto geral, de instruções que podem ser executadas eletronicamente. De forma geral, instruções que podem ser executadas eletro-

nicamente incluem rotinas, programas, objetos, componentes estruturas de dados, etc., que executa, tarefas específicas ou implementam tipos de dados abstratos específicos. A interação de facilitação entre renderizadores de vídeo e acionadores de dispositivos gráficos, conforme descrita em determinadas implementações neste caso, também pode ser concretizada em ambientes de computação distribuídos, onde as tarefas são executadas por dispositivo de processamento ligados de maneira remota, que ficam conectados através de ligações de comunicação e/ou ambiente de rede. Em espacial, em um ambiente de computação distribuída, as instruções que podem ser executadas eletronicamente podem ser localizadas de maneira separada em meios de armazenamento separados, executadas por diferentes processadores e/ou propagadas através de meios de transmissão.

O ambiente de dispositivo eletrônico 700 inclui um dispositivo de computação de finalidade geral, em forma de um computador 702, o qual pode compreender quaisquer dispositivos eletrônicos com capacidades de computação e/ou de processamento. Os componentes do computador 702 podem incluir, mas não se limitar a, uma ou mais unidades de processamento 704, uma memória de sistema 706 e um UBS de sistema 708, que acopla vários componentes de sistemas, incluindo o processador 704 à memória de sistema 706.

UBS de sistema representa um ou mais dentre diversos tipos de estruturas UBS com ou sem fiação, incluindo UBS de memória ou controle de memória, um UBS periférico, uma abertura para gráficos acelerada e um UBS de processamento

ou local, utilizando quaisquer dentre uma variedade de arquiteturas para UBS. À guisa de exemplo, tais arquiteturas podem incluir um UBS de Arquitetura Industrial Padrão - Industrial Standard Architecture BUS - (ISA), um UBS de Arquitetura de Micro Canal - Micro Channel Architecture - (MCA),  
5       tutura de Micro Canal - Micro Channel Architecture - (MCA),  
um UBS aperfeiçoado de ISA (EISA), um UBS local de Associação de Padrões Eletrônicos de Vídeo - Video Electronics Standards Association - (VESA), um UBS de Interconexão de Componente Periférico - peripheral component interconnects -  
10       (PCIM) também conhecido como UBS Mezzanine, algumas misturas destes e assim por diante.

O computador 702 inclui, tipicamente, uma variedade de meios que podem ser acessados eletronicamente. Tais meios podem ser quaisquer meios disponíveis que podem ser  
15       acessados por um computador 702 ou outro dispositivo eletrônico e este inclui tanto um meio volátil, quanto um meio não volátil, um meio removível ou não removível e meios de armazenamento e de transmissão.

A memória de sistema 706 inclui meios de armazenamento eletronicamente acessíveis em forma de uma memória volátil, tal como uma memória de acesso randômico (RAM) 710 e/ou uma memória não volátil, tal como uma memória apenas de leitura (ROM) 712. Um sistema básico de alimentação / saída (BIOS) 714, contendo as rotinas básicas que auxiliam a  
20       transferência de informação entre elementos no interior do computador 702, tal como durante a iniciação, ficam tipicamente armazenados na ROM 712. A RAM 710, tipicamente contém dados e/ou módulos de programas / instruções que podem ser  
25

imediatamente acessados e/ou presentemente operados por uma unidade de processamento 704.

O computador 702 também pode incluir outros meios de armazenamento removíveis / não removíveis e/ou voláteis / não voláteis. À guisa de exemplo, a Figura 7 ilustra um drive ou acionador de disco rígido ou arranjo de acionador de disco 716, para leitura a partir de e escrita para um meio (tipicamente) não removível, não volátil, magnético (não ilustrado separadamente); um acionador de disco 718 para leitura a partir de e escrita para um meio (tipicamente) removível, não volátil, magnético 720 (por exemplo um "floppy disk"); e uma acionador de disco ótico 722 para leitura a partir de e/ou escrita para um disco (tipicamente) removível, não volátil, ótico 724, tal como um CD\_ROM, DVD ou outro meio ótico. O acionador de disco rígido 716, o acionador de disco magnético 718 e o acionador de disco ótico 722 ficam, cada um, conectados a um UBS de sistema 708, através de uma ou mais interfaces de meios de armazenamento 726. Alternativamente, o acionador de disco rígido 716, o acionador de disco magnético 718 e o acionador de disco ótico 722 podem ficar conectados ao UBS de sistema 708 por meio de uma ou mais interfaces separadas ou combinadas (não ilustradas).

Os acionadores dos discos e seus meios eletronicamente acessíveis, associados provêm armazenamento não volátil de instruções eletronicamente executáveis, tais como estruturas de dados, módulos de programas e outros dados para computadores 702. Embora o exemplo de computador 702 illustre um disco rígido 716, um disco magnético removível 720

e um disco ótico removível 724, deve ser considerado que outros tipos de meios que podem ser acessados eletronicamente podem armazenar instruções que podem ser acessados por um dispositivo eletrônico, tais como K7s magnéticos ou outros dispositivos de armazenamento eletrônico, memória flash, CD - ROM, discos digitais versáteis (DVDs) ou outro armazenamento ótico, RAM, ROM, memórias de leitura apenas, programáveis, que podem ser apagadas eletricamente (EEPROM) e assim por diante. Tais meios também podem incluir os assim denominados chips de circuito integrado (IC), de fiação rígida ou finalidade especial. Em outras palavras, qualquer meio eletronicamente acessível pode ser utilizado para concretização do meio de armazenamento do exemplo de sistema e ambiente eletrônico 700.

Qualquer quantidade de módulos de programas (ou outras unidades ou conjuntos de instruções) pode ser armazenada em disco rígido 716, disco magnético 720, disco ótico 724, ROM 712 e/ou RAM 710, incluindo, à guisa de exemplo geral, um sistema operacional 728, um ou mais programas de aplicação 730, outros módulos de programas 732 e dados de programas 734. À guisa de exemplo específico, mas não de forma limitante, um renderizador de vídeo 410, uma interface gráfica 412 e uma interface de acionador de dispositivo 414 (todos da Figura 4) podem constituir parte do sistema operacional 728. O acionador de dispositivo gráfico 422 pode constituir parte dos módulos de programa 732, opcionalmente com uma ligação próxima e/ou relacionador integral com o sistema operacional 728. Do mesmo modo, um programa de instigação,

tal como Windows<sup>®</sup> Media<sup>®</sup> 9, tratando-se de um exemplo de programa de aplicação 730. O controle de imagem e/ou dados gráficos que correntemente se encontram na memória de sistema podem constituir parte dos dados de programa 734.

5 Um usuários que se encontra modificando ProcAmp ou outros ajustes de vídeo, por exemplo, pode dar entrada a um comando e/o informação ao computador 702, através de dispositivos de alimentação de entrada, tais como um teclado 736 e um dispositivo de apontamento 738 (por exemplo, um  
10 "mouse"). Outros dispositivos de alimentação de entrada 740 (não especificamente ilustrados) podem incluir um microfone, joystick, coxim de jogo, antena de satélite, entrada em série, scanner e/ou similares. Estes e outros dispositivos de alimentação de entrada ficam conectados à unidade de proces-  
15 samento 704, através de interfaces de alimentação de entrada / alimentação de saída 742, que ficam acopladas a UBS de sistema 708. No entanto, estes e o/ou dispositivos de alimentação de saída podem, ao invés disso, ficarem conectados por outras estruturas de interface e UBS, tais como entradas  
20 em paralelo, abertura para jogos, UBS em série universal (USB), uma interface IEEE 1394 ("fiação de fogo" - "Firewire"), uma interface sem fio IEEE 802.11, uma interface sem fio Bluetooth<sup>®</sup> e assim por diante.

Um monitor / tela de exibição 744 (a qual se tra-  
25 ta de um exemplo de dispositivo de exibição 436 da Figura 4) ou outro tipo de dispositivo de exibição também podem ficar conectados ao UBS de sistema 708, através de uma interface, tal como um adaptador para vídeo 746. O adaptador para vídeo

746 (ou outro componente) pode ser ou incluir uma placa gráfica (a qual se trata de um exemplo de dispositivo gráfico 424), para processamento de cálculos de intensidade gráfica e para manipulação de exigências de demanda de exibição.

5 Tipicamente, uma placa gráfica inclui um GPU (tal como um GPU 426), vídeo RAM (VRAM) (o qual se trata de um exemplo de memória de vídeo 432), etc., para facilitação de realização de expedição de operações gráficas. Em adição ao monitor 744, outros dispositivos periféricos de alimentação de saída

10 podem incluir componentes tais como alto-falantes (não ilustrados) e uma impressora 748, a qual pode ficar conectada ao computador 702, através de interfaces de alimentação de entrada / alimentação de saída 742.

O computador 402 pode operar em um ambiente de

15 rede, fazendo uso de conexões lógicas a um ou mais computadores remotos, tais como um dispositivo de computação remota 750. À guisa de exemplo, o dispositivo de computação remota 750 pode ser um computador pessoal (microcomputador), um computador portátil (por exemplo, computador do tipo laptop,

20 computador de mesa, PDA, estação móvel, etc.), um computador de mão ou de dimensionado para bolso, um dispositivo para jogos, um servidor, um direcionador, um computador de rede, um dispositivo igualado, outro nóculo de rede comum ou outro tipo de computador, conforme listado acima e assim por dian-

25 te. No entanto, o dispositivo de computação remota 750, se encontra ilustrado como computador portátil, que pode incluir muitos ou todos dentre os elementos e características descritos neste caso, em relação ao computador 702.

Conexões lógicas entre o computador 702 e o computador remoto 750 se encontram ilustradas como rede de área local (LAN) 745 e uma rede de ampla área, geral (WAN) 754. Tais ambientes de rede são lugar comum em escritórios, redes de computadores de amplitude de empreendimentos, redes de computadores, intranets, Internet, redes de telefonia fixa e móveis, outras redes sem fio, redes de jogos, algumas misturas destes e assim por diante.

Quando implementado em um ambiente de rede LAN, o computador 702, geralmente, fica conectado à LAN 752 através de uma interface ou adaptação de rede 756. Quando implementado em um ambiente de rede WAN, o computador 702, tipicamente, inclui um modem 758 ou outros meios de estabelecimento de comunicação através de WAN 754. O modem 758, o qual pode ser interno ou externo ao computador 702 pode ficar conectado ao UBS de sistema 708, através de interfaces de alimentação de entrada / alimentação de saída 742 ou qualquer outro(s) meio(s) apropriado. Deve ser observado que as conexões de rede ilustradas tratam-se de exemplos e que outros meios de estabelecimento de ligação(ões) de comunicação, entre computadores 702 e 750 podem ser empregados.

Em um ambiente de rede, tal como aquele ilustrado com o ambiente de dispositivo eletrônico 700, módulos de programas ou outras instruções que se encontram ilustradas com relação ao computador 702 ou partes do mesmo, podem ser completa ou parcialmente armazenados em um dispositivo de armazenamento de memória remoto. À guisa de exemplo, programas de aplicação remota 760 residem em um componente de me-

mória de computador remoto 750, contudo podem ser utilizados ou acessados de alguma maneira através do computador 702. Do mesmo modo, para fins de ilustração, programas de aplicação 730 e outras instruções que podem ser executadas eletronicamente, tais como um sistema operacional 728 se encontram i-  
5 lustradas neste caso, como blocos discretos, mas reconhece-se que tais programas, componentes e outras instruções residem, em vários instantes em diferentes componentes de armazenamento do dispositivo de computação 702 (e/ou dispositivo  
10 de computação remoto 750) e podem ser executados pelo(s) processador(es) de dados 704 do computador 702 (e/ou aqueles do dispositivo de computação remota 750).

Embora sistemas, meios, métodos, protocolos, soluções, processos, disposições e outras implementações forma  
15 descritos em linguagem específica em relação a características e/ou diagramas estruturais, lógicos, algorítmicos e funcionais, deve ser entendido que a invenção, definida nas reivindicações em anexo não se limita, necessariamente, às características ou diagramas específicos, descritos. Ao con-  
20 trário, as característica e diagramas específicos encontram-se divulgados em forma de exemplos de implementação da invenção reivindicada.

## REIVINDICAÇÕES

1. Método implementado por computador para facilitar interação entre um ou mais renderizadores de vídeo (410) e pelo menos um acionador de dispositivo gráfico (422), o método **CARACTERIZADO** pelo fato de que compreende as etapas de:

indagar (602), por meio de um renderizador de vídeo (410) dentre um ou mais renderizadores de vídeo (410), ao pelo menos um acionador de dispositivo gráfico (422), com referênciã às capacidades de propriedade de controle de amplificação de processo (ProcAmp), a indagação incluindo uma descrição de vídeo para ser exibido;

gerar, com base na descrição de vídeo para ser exibido, uma resposta incluindo informação relativa a capacidades de propriedade de controle de ProcAmp;

enviar (620) a resposta para o renderizador de vídeo (410) a partir do pelo menos um acionador de dispositivo gráfico (422), em que a informação relativa às capacidades de propriedade de controle de ProcAmp inclui, pelo menos, um subconjunto de capacidades de propriedade de controle de ProcAmp que o pelo menos um acionador de dispositivo gráfico (422) pode oferecer para o renderizador de vídeo (410);

proporcionar (622) ao renderizador de vídeo (410) a partir do acionador de dispositivo gráfico (422), ajustes ProcAmp que podem ser executados simultaneamente com as operações de processamento de vídeo associadas com o subconjunto de propriedades de controle de ProcAmp;

selecionar (608), no renderizador de vídeo (410),

uma propriedade de controle de ProcAmp associada com um dos subconjuntos de capacidades de propriedade de controle de ProcAmp;

5 receber (610) no acionador de dispositivo gráfico (422), a partir do renderizador de vídeo (410), uma requisição de valores associados com a propriedade do controle de ProcAmp selecionada;

proporcionar (624) valores para o renderizador de vídeo (410) em resposta à requisição de valores associados com a propriedade do controle de ProcAmp selecionada, os valores sendo usados em operações de processamento de vídeo;

receber (614) no acionador de dispositivo gráfico (422), a partir do renderizador de vídeo (410), um comando para abrir um objeto em feixe ProcAmp;

15 abrir (626) o objeto em feixe ProcAmp;

enviar para o renderizador de vídeo (410), a partir do acionador de dispositivo gráfico (422), outra resposta apresentando um manuseio que aponta para o objeto em feixe ProcAmp aberto;

20 receber (616) no acionador de dispositivo gráfico (422), a partir do renderizador de vídeo (410), um comando para executar um ajuste ProcAmp; e

causar (628) desempenho do ajuste ProcAmp, em que:

25 o comando inclui uma instrução para executar uma operação de processamento de vídeo (302) associada com a propriedade de controle de ProcAmp selecionada simultaneamente ao ajuste ProcAmp;

causar desempenho do ajuste ProcAmp compreende

causar um desempenho simultâneo do ajuste ProcAmp e a operação de processamento de vídeo (302); e

5 a informação relativa às capacidades de propriedade controle de ProcAmp é direcionada para operações de processamento de vídeo que são executadas simultaneamente com os ajustes ProcAmp.

2. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o pelo menos um conjunto de capacidades de ProcAmp, que o pelo menos um acionador de dispositivo gráfico (422) pode oferecer ao renderizador de vídeo (410), compreendem capacidades de ProcAmp alocadas ao renderizador de vídeo (410) a partir do pelo menos um acionador de dispositivo gráfico (422).

3. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o acionador de dispositivo gráfico (422) tem um ou mais conjuntos de capacidades de propriedade de controle de ProcAmp ajustados para uma ou mais descrições correspondendo à descrição de vídeo para ser exibido.

20 4. Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que compreende ainda as etapas de:

enviar (618) um comando para fechamento de um objeto em feixe de processamento de vídeo ao acionador de dispositivo gráfico (422) a partir do renderizador de vídeo (410);

receber um comando a partir do renderizador de vídeo (410) no acionador de dispositivo gráfico (422); e

fechar (630), pelo acionador de dispositivo gráfico (422), o objeto em feixe de processamento de vídeo.

5           5. Meio eletronicamente acessível, compreendendo instruções eletronicamente executáveis, as quais quando executadas, provocam ações, **CARACTERIZADO** pelo fato de que as ditas ações compreendem as etapas de:

                  receber (602) uma indagação em um acionador de dispositivo gráfico (422), a partir de um renderizador de vídeo (410), a indagação compreendendo:

10                   uma requisição de informação referente à capacidades de propriedade de controle de amplificação de processo (ProcAmp); e

                  uma descrição do tipo de vídeo para ser exibido;

15                   gerar, com base na descrição de vídeo para ser exibido, uma resposta incluindo a informação requisitada relativa a capacidades de propriedade de controle de ProcAmp;

                  enviar (620) a resposta para o renderizador de vídeo (410) a partir do acionador de dispositivo gráfico (422);

20                   proporcionar (622) ao renderizador de vídeo (410) a partir do acionador de dispositivo gráfico (422), ajustes ProcAmp que podem ser executados simultaneamente com as operações de processamento de vídeo associadas com as propriedades de controle de ProcAmp;

25                   selecionar (608), no renderizador de vídeo (410), uma propriedade de controle de ProcAmp associada com uma das capacidades de propriedade de controle de ProcAmp;

receber (610) no acionador de dispositivo gráfico (422), a partir do renderizador de vídeo (410), uma requisição de valores associados com a propriedade do controle de ProcAmp selecionada;

5           proporcionar (624) valores para o renderizador de vídeo (410) em resposta à requisição de valores associados com a propriedade do controle de ProcAmp selecionada, os valores sendo usados em operações de processamento de vídeo;

          receber (614) no acionador de dispositivo gráfico (422), a partir do renderizador de vídeo (410), um comando para abrir um objeto em feixe ProcAmp;

          abrir (626) o objeto em feixe ProcAmp;

          enviar para o renderizador de vídeo (410), a partir do acionador de dispositivo gráfico (422), outra resposta apresentando um manuseio que aponta para o objeto em feixe ProcAmp aberto;

          receber (616) no acionador de dispositivo gráfico (422), a partir do renderizador de vídeo (410), um comando para executar um ajuste ProcAmp; e

20           causar (628) desempenho do ajuste ProcAmp, em que:

          o comando inclui uma instrução para executar uma operação de processamento de vídeo (302) associada com a propriedade de controle de ProcAmp selecionada simultaneamente ao ajuste ProcAmp;

25           causar desempenho do ajuste ProcAmp compreende causar um desempenho simultâneo do ajuste ProcAmp e a operação de processamento de vídeo (302); e

          a informação relativa às capacidades de pro-

priedade controle de ProcAmp é direcionada para operações de processamento de vídeo que são executadas simultaneamente com os ajustes ProcAmp.

5 6. Meio eletronicamente acessível de acordo com a reivindicação 5, **CARACTERIZADO** pelo fato de que as uma ou mais operações de processamento de vídeo desempenhadas simultaneamente com ajustes ProcAmp inclui pelo menos uma operação de processamento de vídeo (302), selecionada dentre um grupo compreendendo: operação de desinterlace, correções de  
10 alcance de aspecto, conversões de espaço de cor, conversões de alcance de estrutura, operações de espelhamento vertical ou horizontal, operações de combinações alfa e nenhuma.

7. Meio eletronicamente acessível de acordo com a reivindicação 5, **CARACTERIZADO** pelo fato de que o comando é  
15 direcionado a uma transferência de blocos de bits.

8. Meio eletronicamente acessível, de acordo com a reivindicação 7, **CARACTERIZADO** pelo fato de que o comando instrui o acionador de dispositivo gráfico (422) para fins da concretização da transferência de blocos de bits ocorrer  
20 em conjunto com o desempenho do ajuste ProcAmp no vídeo.

9. Meio eletronicamente acessível, de acordo com a reivindicação 7, **CARACTERIZADO** pelo fato de que o comando instrui o acionador de dispositivo gráfico (422) para fins da concretização da transferência de blocos de bits ocorre  
25 em conjunto com o desempenho do ajuste ProcAmp para geração de uma imagem para a totalidade da tela de um dispositivo de exibição, esta uma imagem armazenada na memória como um bloco único e incluindo uma estrutura do vídeo após o ajuste

ProcAmp.

10. Meio eletronicamente acessível, de acordo com a reivindicação 5, **CARACTERIZADO** pelo fato de que o comando estipula pelo menos um valor de propriedade de controle ou  
5 modificação para um valor de propriedade de controle para o ajuste ProcAmp no vídeo.

11. Meio eletronicamente acessível, de acordo com a reivindicação 5, **CARACTERIZADO** pelo fato de que compreende instruções eletronicamente executáveis que, quando execu-  
10 das precipitam uma ação adicional, compreendendo:

receber, no renderizador de vídeo (410), uma requisição de operação a partir de uma aplicação de instigação, esta requisição de operação incluindo uma requisição para executar um ajuste ProcAmp no vídeo.

12. Meio eletronicamente acessível, de acordo com a reivindicação 5, **CARACTERIZADO** pelo fato de que compreendem instruções eletronicamente executáveis que, quando executadas precipitam uma ação adicional, compreendendo:

emitir um comando a partir do renderizador de vídeo (410) em direção ao acionador de dispositivo gráfico (422) para fechamento de um objeto em feixe ProcAmp.

13. Meio eletronicamente acessível, de acordo com a reivindicação 5, **CARACTERIZADO** pelo fato de que as ações de receber uma indagação e enviar a resposta são efetuadas,  
25 pelo menos parcialmente, por meio da propagação da indagação e da resposta entre o renderizador de vídeo (410) e o acionador de dispositivo gráfico (422) com a utilização de pelo menos uma interface gráfica (412) e uma interface de aciona-

dor de dispositivo (414).

14. Meio eletronicamente acessível, de acordo com a reivindicação 5, **CARACTERIZADO** pelo fato de que pelo menos uma porção das instruções eletronicamente executáveis compreende pelo menos parte de um sistema de operação.

15. Meio eletronicamente acessível, de acordo com a reivindicação 5, **CARACTERIZADO** pelo fato de que pelo menos uma porção das instruções eletronicamente executáveis compreende pelo menos parte de um acionador de dispositivo gráfico (422).

16. Meio eletronicamente acessível, de acordo com a reivindicação 5, **CARACTERIZADO** pelo fato de que as propriedades de controle ProcAmp são selecionadas dentre um grupo compreendendo: nenhuma, brilho, contraste, matiz e saturação.

17. Meio eletronicamente acessível, de acordo com a reivindicação 5, **CARACTERIZADO** pelo fato de que os valores são selecionados dentre um grupo, compreendendo: alcance, máximo, mínimo, dimensão de etapa / incremento e falho.

18. Meio eletronicamente acessível, de acordo com a reivindicação 5, **CARACTERIZADO** pelo fato de que a uma ou mais operações de processamento de vídeo que podem ser executadas simultaneamente com ajustes ProcAmp incluem pelo menos uma operação de processamento de vídeo (302) selecionada a partir de uma pluralidade de operações de processamento de vídeo compreendendo: uma operação de conversão YUV para RGB, uma operação de extensão X, uma operação de extensão Y, uma operação de ajuste de subregião retangular, uma operação de

combinação alfa, e nenhuma.

19. Sistema para facilitar interação entre um renderizador de vídeo (410) e um acionador de dispositivo gráfico (422), o sistema **CARACTERIZADO** pelo fato de que compreende:

um processador;

lógica para renderização de vídeo adaptada para preparo de indagações que requisitam informações referentes às capacidades de amplificação de processo (ProcAmp), que podem ser aplicadas a vídeo para ser exibido, em que as indagações incluem uma descrição do tipo de vídeo que é para ser exibido;

lógica para acionador de dispositivo gráfico (422) adaptada para preparar respostas que indicam quais capacidades de ProcAmp podem ser aplicadas a vídeo para ser exibido com base na descrição do tipo de vídeo para ser exibido; e

lógica de interface que é adaptada para facilitar interação entre a lógica para renderização de vídeo e a lógica para acionador de dispositivo gráfico (422), a interação compreendendo:

receber uma indagação (602) em um acionador de dispositivo gráfico (422), a partir de um renderizador de vídeo (410), a indagação compreendendo:

uma requisição de informação referente à capacidades de propriedade de controle de amplificação de processo (ProcAmp); e

a descrição do tipo de vídeo para ser exibido;

gerar, com base na descrição de vídeo para ser exibido, uma resposta incluindo a informação requisitada relativa a capacidades de propriedade de controle de ProcAmp;

enviar (620) a resposta para o renderizador de vídeo (410) a partir do acionador de dispositivo gráfico (422);

proporcionar (622) ao renderizador de vídeo (410) a partir do acionador de dispositivo gráfico (422), ajustes ProcAmp que podem ser executados simultaneamente com as operações de processamento de vídeo associadas com as propriedades de controle de ProcAmp;

selecionar (608), no renderizador de vídeo (410), uma propriedade de controle de ProcAmp associada com uma das capacidades de propriedade de controle de ProcAmp;

receber (610) no acionador de dispositivo gráfico (422), a partir do renderizador de vídeo (410), uma requisição de valores associados com a propriedade do controle de ProcAmp selecionada;

proporcionar (624) valores para o renderizador de vídeo (410) em resposta à requisição de valores associados com a propriedade do controle de ProcAmp selecionada, os valores sendo usados em operações de processamento de vídeo;

receber (614) no acionador de dispositivo gráfico (422), a partir do renderizador de vídeo (410), um comando para abrir um objeto em feixe ProcAmp;

abrir (626) o objeto em feixe ProcAmp;

enviar para o renderizador de vídeo (410), a partir do acionador de dispositivo gráfico (422), outra respos-

ta apresentando um manuseio que aponta para o objeto em fei-  
xe ProcAmp aberto;

receber (616) no acionador de dispositivo gráfico  
(422), a partir do renderizador de vídeo (410), um comando  
5 para executar um ajuste ProcAmp; e

causar (628) desempenho do ajuste ProcAmp, em que:

o comando inclui uma instrução para executar  
uma operação de processamento de vídeo (302) asso-  
ciada com a propriedade de controle de ProcAmp se-  
10 lecionada simultaneamente ao ajuste ProcAmp;

causar desempenho do ajuste ProcAmp compreende  
causar um desempenho simultâneo do ajuste ProcAmp  
e a operação de processamento de vídeo (302); e

a informação relativa às capacidades de proprieda-  
15 de controle de ProcAmp é direcionada para operações de pro-  
cessamento de vídeo que são executadas simultaneamente com  
os ajustes ProcAmp.

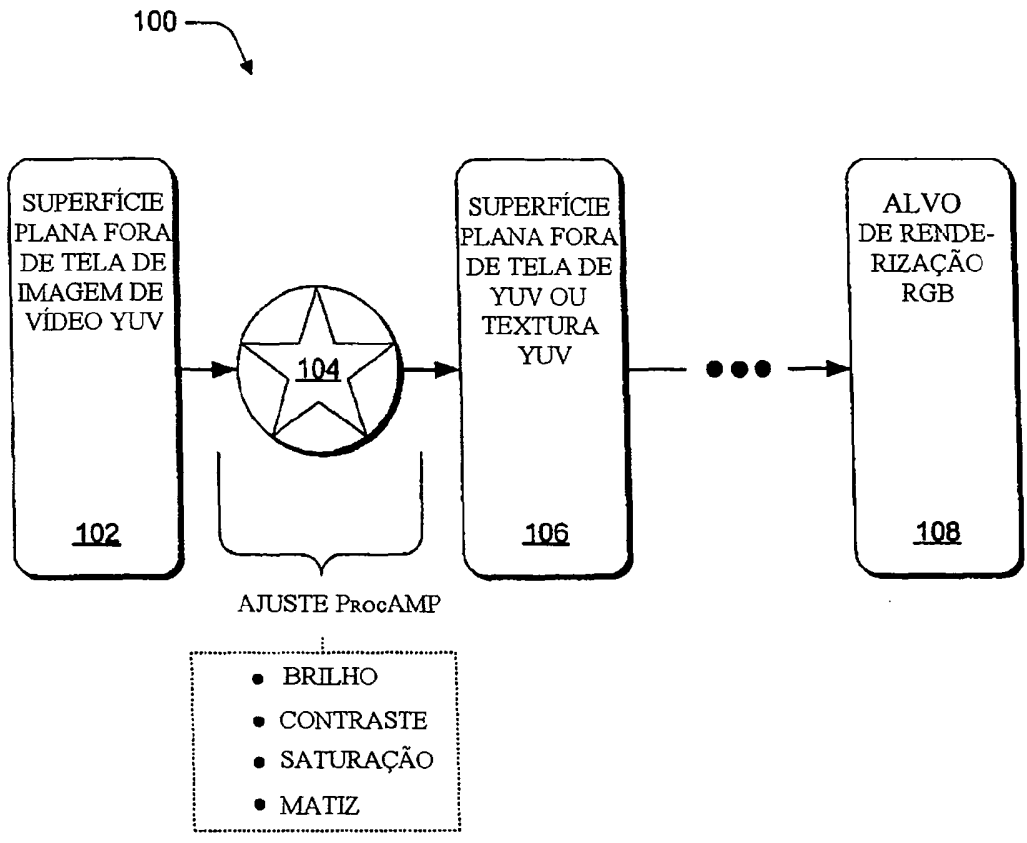


Fig. 1

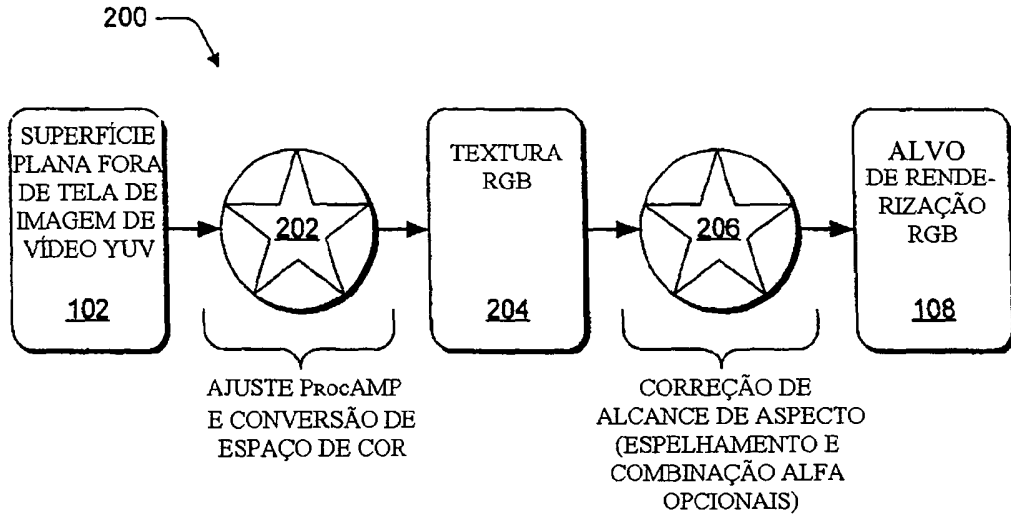


Fig. 2

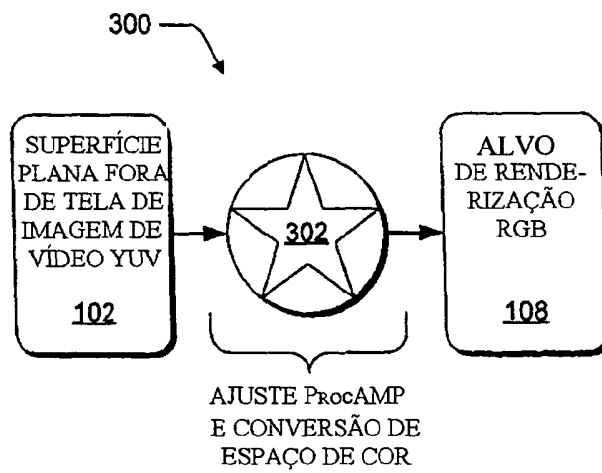


Fig. 3

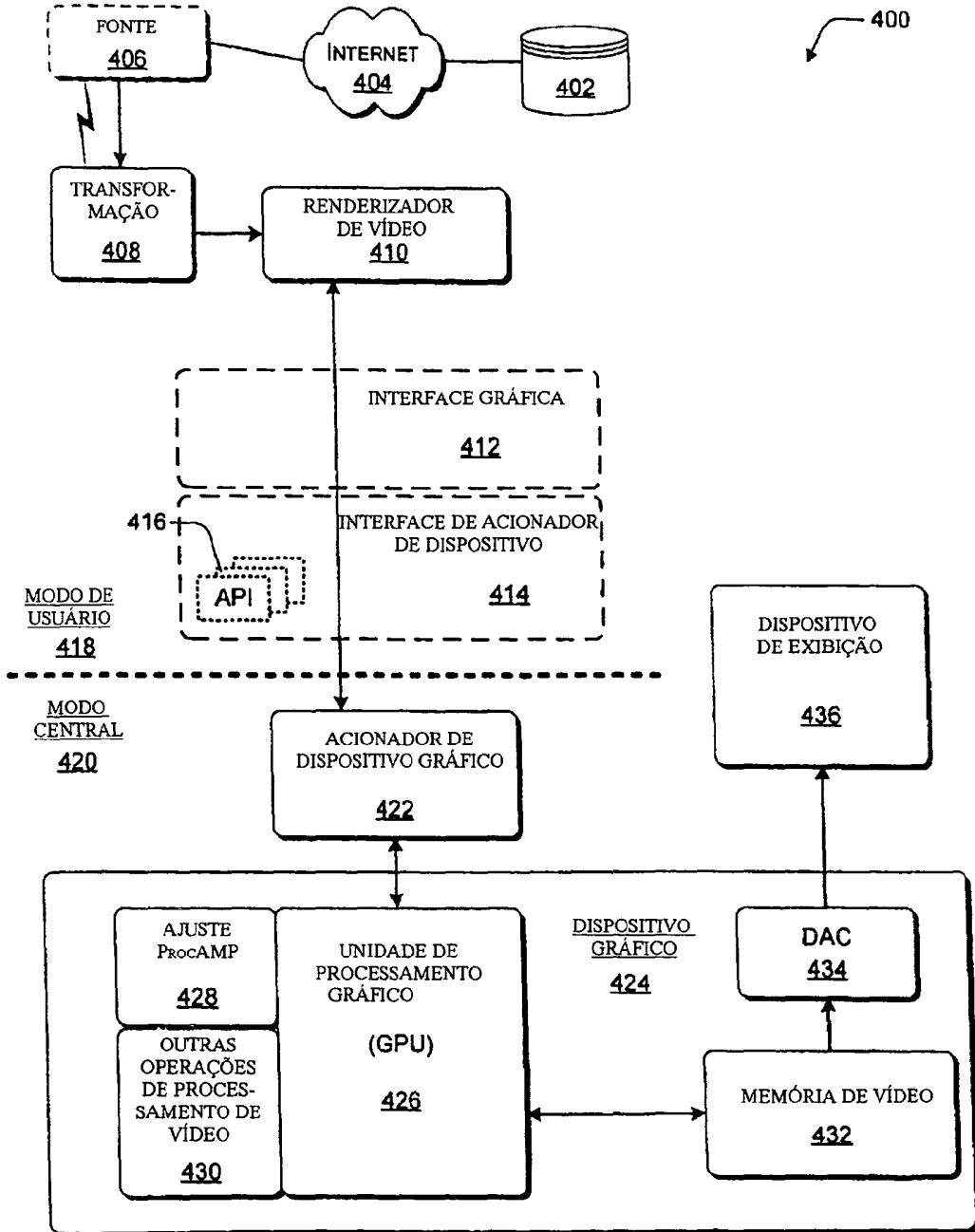


Fig. 4

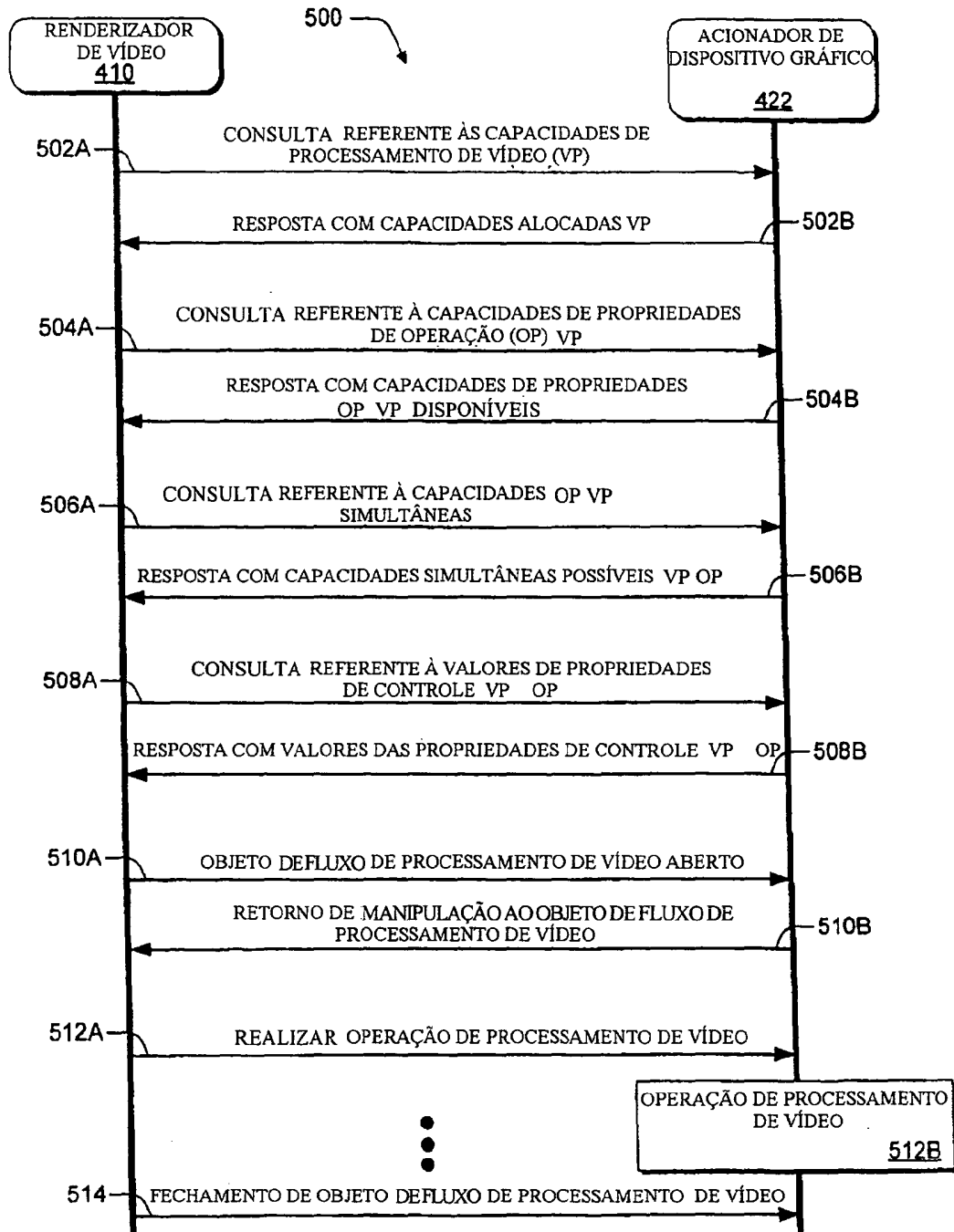
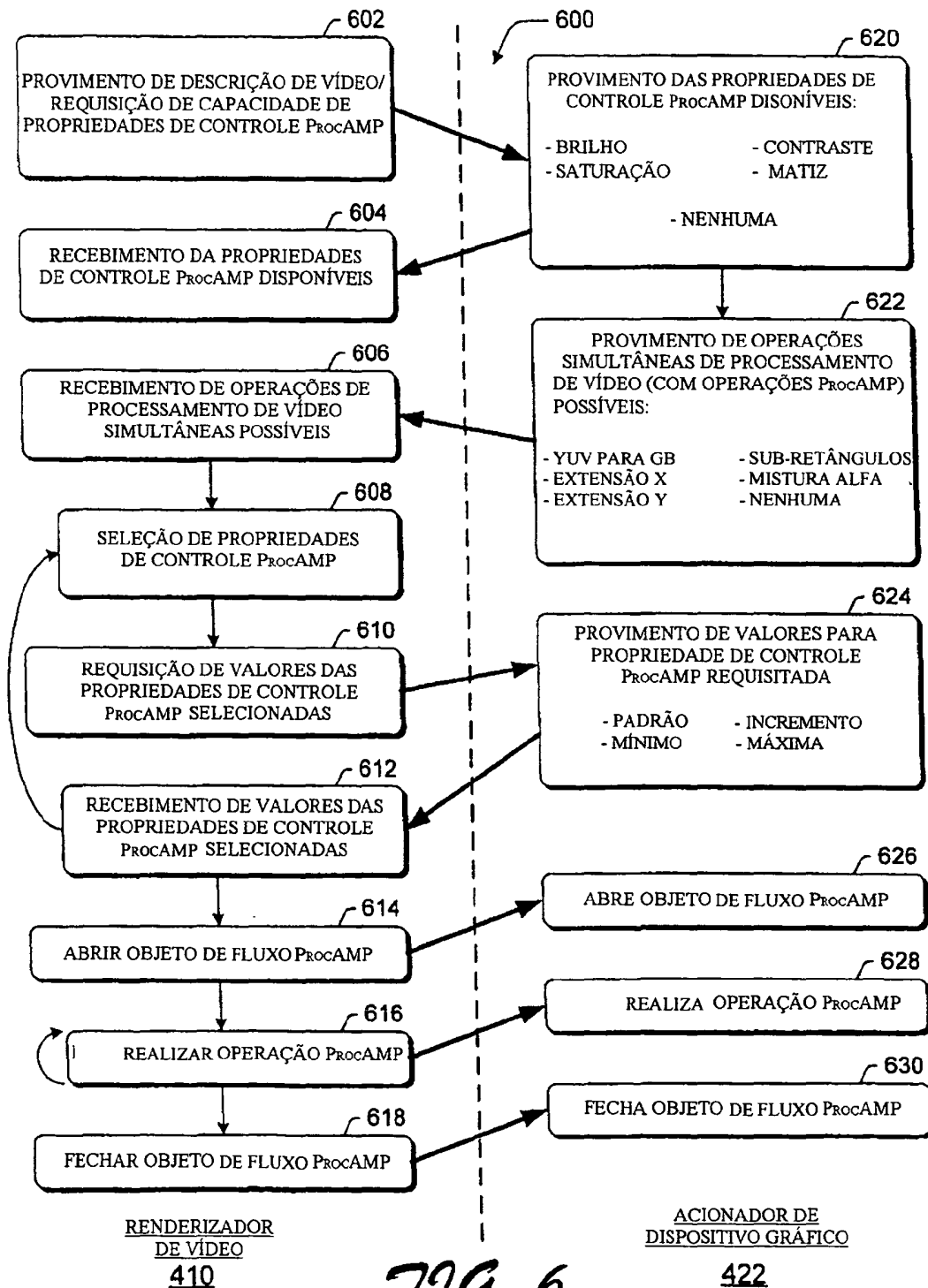
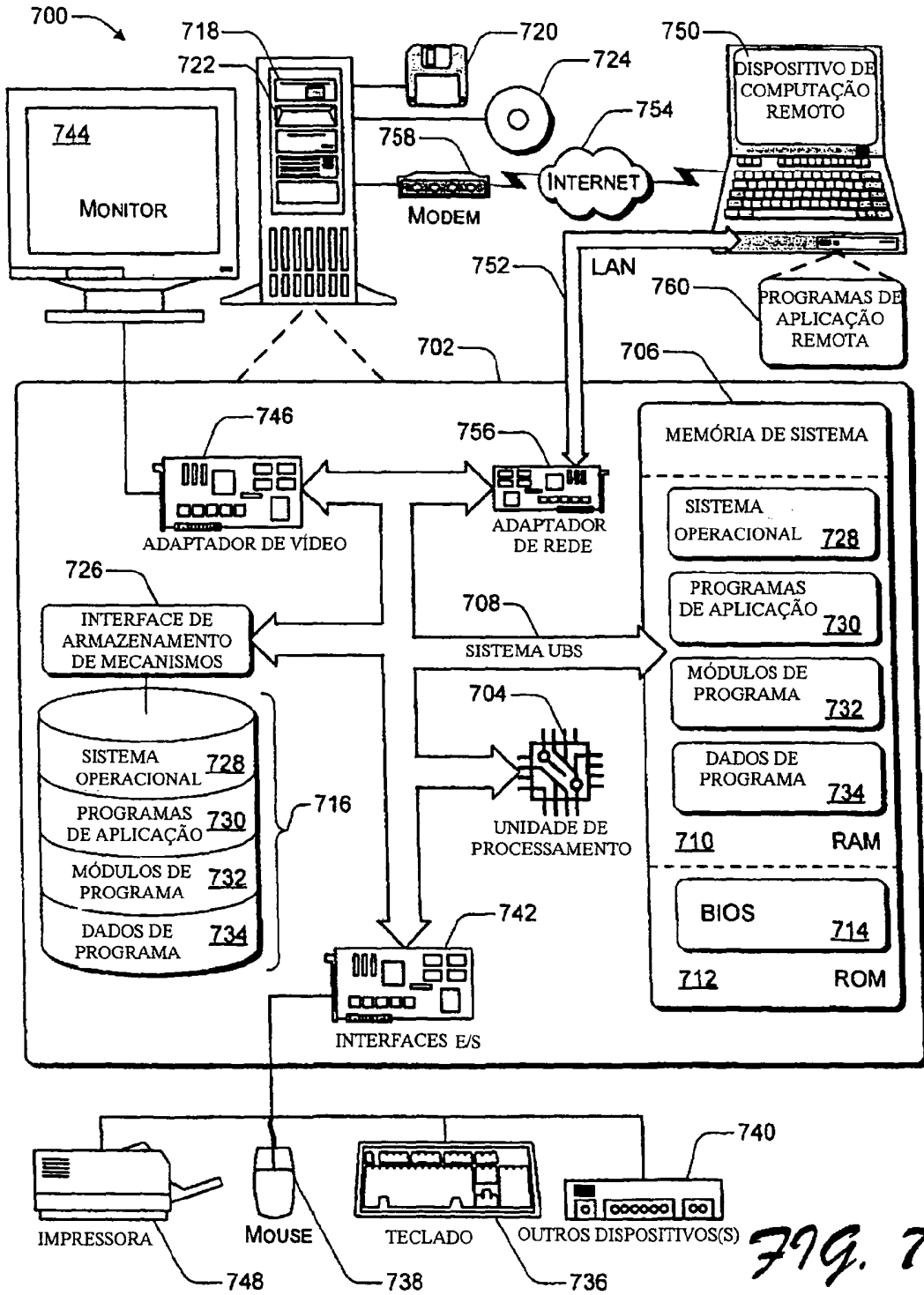


Fig. 5





## RESUMO

### **"MÉTODO E SISTEMA PARA FACILITAR INTERAÇÃO ENTRE UM RENDERIZADOR DE VÍDEO E UM ACIONADOR DE DISPOSITIVO GRÁFICO E MEIO ELETRONICAMENTE ACESSÍVEL".**

5           A facilitação da interação pode ser possibilitada através de protocolos de comunicação e/ou APIs, que permitem que informações referentes às capacidades de processamento de imagem do hardware gráfico associado possam ser trocadas entre acionador de dispositivos gráficos e renderizadores de vídeo (410). Em um primeiro exemplo de implementação de mecanismo, instruções que podem ser eletronicamente executadas do mesmo, para um renderizador de vídeo (410), precipitam ações incluindo: lançamento de uma indagação a partir do renderizador de vídeo (410), em direção ao acionador de dispositivo gráfico (422), esta indagação requisitando informações referentes às capacidades de amplificação do processo (ProcAmp); e recebimento de resposta no renderizador de vídeo (410), a partir do acionador de dispositivo gráfico (422), esta resposta incluindo a informação requisitada referente à capacidades ProcAmp. Em um segundo exemplo de implementação de mecanismo, um acionamento de dispositivo gráfico precipita ações incluindo: recebimento de uma indagação no acionador de dispositivo gráfico (422), a partir de um renderizador de vídeo (410), esta indagação requisitando informações referentes às capacidades ProcAmp; e envio de uma resposta ao renderizador de vídeo (410), a partir do acionador de dispositivo gráfico (422), esta resposta incluindo a informação requisitada que se refere às capacidades ProcAmp.