



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

(11) PI 0410112-0 B1

(22) Data do Depósito: 07/05/2004

(45) Data de Concessão: 17/01/2017



(54) Título: MÉTODO PARA MAPEAMENTO DE DADOS DE EXIBIÇÃO

(51) Int.Cl.: G06F 3/14; G06F 9/44; G06F 9/54; G09G 5/14; G09G 5/39

(52) CPC: G06F 3/1454, G06F 9/4445, G06F 9/542, G09G 5/14, G06F 2209/544, G06F 2209/545, G09G 5/39

(30) Prioridade Unionista: 07/05/2003 US 10/249,781

(73) Titular(es): INTERNATIONAL BUSINESS MACHINES CORPORATION

(72) Inventor(es): JOSEPH M. CRICHTON

Relatório Descritivo da Patente de Invenção para: **"MÉTODO PARA MAPEAMENTO DE DADOS DE EXIBIÇÃO"**.

Campo Técnico

[0001] Essa invenção se refere ao mapeamento de dados de exibição, e mais particularmente ao mapeamento de dados de exibição de um ou mais formatos para um formato compartilhado.

Técnica Anterior

[0002] A colaboração entre indivíduos assume muitas formas como face-a-face, chamadas telefônicas e e-mails sendo alguns dos mais ubíquos. À medida que a tecnologia tem progredido, os indivíduos têm procurado colaborar ainda mais através de compartilhamento de vídeo, voz, marcações em quadro, chat, bem como aplicativos de computador. Dois produtos de software de colaboração comum que permitem tal compartilhamento são o LOTUS SAMETIME® e o Microsoft NetMeeting®. Um problema com muitos dos produtos de software atuais é a dependência de plataforma dos mesmos. Por exemplo, o Microsoft NetMeeting® só trabalha em sistemas que usam uma versão do sistema operacional Microsoft Windows®. Plataformas diferentes gerenciam dados de exibição de maneira diferente. Isso apresenta um obstáculo significativo para criar produtos de colaboração que podem ser usados através de plataformas.

[0003] Por exemplo, dados de exibição incluem dados de pixel. Dados de pixel incluem informação sobre a localização do pixel, e a profundidade do pixel. Vários formatos são usados para representar a informação de cor em dados de pixel. Consequentemente, a fim de implementar o compartilhamento de dados de exibição através de plataformas, os dados de pixel podem ser convertidos para um formato compartilhado. Entretanto, alguns fatores impedem muitas implementações de converter, de forma eficiente, informações de pixel para um formato compartilhado. Um fator é a necessidade de determinar a janela que "detém" cada pixel na área de exibição a ser convertida. Uma janela detém um pixel quando ela fornece informação de pixel para o pixel.

[0004] Em um ambiente de exibição de janelas típico, as janelas podem ter uma relação hierárquica. Por exemplo, uma janela pai pode ser criada que inclui dentro do seu espaço de exibição uma ou mais janelas filhas. As janelas filhas podem ter também uma ou mais janelas filhas de suas propriedades. Cada janela ocupa uma porção do espaço de exibição de qualquer janela antecessora (isto é, janela pai, janela avó, etc.). Janelas irmãs ou compartilham a mesma janela pai ou não possuem janela pai (isto é, elas são exibidas na área de trabalho). Janelas irmãs são atribuídas em uma ordem de empilhamento. A ordem de empilhamento

determina a ordem na qual as janelas irmãs são arrastadas, e como um resultado que janela irmã está "no topo", quando as áreas de exibição para duas janelas se sobrepõem. O tamanho, a ordem de empilhamento e o número de janelas são alterados frequentemente pelo usuário. Por exemplo, uma janela pai pode ser maximizada para ocupar uma área de exibição inteira. Além disso, um usuário pode selecionar uma janela parcialmente atrás de uma janela irmã, resultando em uma janela selecionada sendo mostrada no topo de uma janela irmã.

[0005] A determinação da propriedade do pixel é importante em um sistema Windows X, uma vez que os dados de pixel podem ser formatados de forma diferente para cada janela nesse sistema. O sistema Windows X é um sistema de janela cliente-servidor, no qual um "cliente X" (aplicativo) realiza processamento que inclui comandos para alterar uma exibição. Esses comandos podem ser fornecidos para um "servidor X" que implementa a alteração de exibição (isto é, "serve" a imagem para um usuário). O servidor X é residente em um computador com o display, enquanto o cliente X pode estar residente em qualquer computador em uma rede de computadores que inclui o computador com o display.

[0006] Os formatos de exibição típicos variam em "profundidade", isto é, o número de bits usados para os dados

de pixel para cada pixel e "visual", ou seja, como os dados de pixel são interpretados. A profundidade determina o número de cores possíveis que podem ser exibidas de uma vez por pixel dentro de uma janela. Por exemplo, os dados de pixel tendo uma profundidade de oito bits permitem que até duzentas e cinquenta e seis cores (28) sejam exibidas simultaneamente. Em geral, o visual determina se os dados de pixel devem ser interpretados como incluindo os valores de cores ou como incluindo um ou mais indexes em tabelas de cores que contém os valores de cores. Existem seis tipos de padrão de visuais em um ambiente de sistema Windows X: dados de pixel de cores verdadeira incluem os valores de cores vermelho-verde-azul (RGB - *Red-Green-Blue*) codificados em dados de pixel, dados de pixel de cinza estático (*StaticGray*) e cor estática (*StaticColor*) contém um índice em uma tabela de cores contendo valores de cor não alteráveis, dados de pixel de cor direta (*DirectColor*) incluem três valores de índice para procurar os valores de cor RGB em três tabelas de cores modificável separadas, e dados de pixel de escala de cinza (*GrayScale*) e Pseudocor (*PseudoColor*) compreendem um índice em uma tabela de cor modificável que contém os valores de cor. Os três visuais finais permitem que os valores na uma ou mais tabelas de cores sejam modificados, permitindo, desse modo, que as cores reais exibidas para um valor particular

sejam variáveis. Um formato de vinte quatro bits de cor verdadeira é um formato comumente usado para dados de exibição. Com esse formato, o valor real para cada cor (isto é, vermelho, verde e azul) é representado por meio de uma única porção de oito bits do valor de vinte e quatro bits. Numerosos sistemas e aplicações são configurados para suportar esse formato. Por exemplo, a linguagem de programação desenvolvida pela Sun Microsystems suporta o formato de vinte e quatro bits cor verdadeira e tem sido implementado em numerosos sistemas e plataformas.

[0007] Determinar quando uma área do display foi modificada é um outro fator que impede a conversão de dados de exibição de forma eficiente. Por exemplo, um servidor X fornece um fluxo de evento baseado em exibição e mecanismo de questionamento para informar um aplicativo de um evento inicializado pelo usuário, permitindo, desse modo, que o aplicativo interaja com o usuário. Um aplicativo pode especificar quais eventos ele deseja ser notificado a respeito, e tomar a ação apropriada com base no evento. Eventos comuns incluem a criação e destruição de uma janela, a remodelação da janela, a alteração da ordem de empilhamento de uma janela, etc. Entretanto, um servidor X não provê um evento que sinaliza quando uma área de um display foi modificada. Consequentemente, de forma a compartilhar uma

área de exibição com outro sistema, os dados de exibição para a área total do display têm que ser continuamente copiados e monitorados.

[0008] Algumas abordagens têm fornecido uma implementação de compartilhamento de aplicativo incluindo o compartilhamento de dados de exibição em um sistema Windows X. Por exemplo, um programa de visualização separado pode ser executado. Essa abordagem é usada na solução de computação de rede virtual (VCN - *Virtual Network Computing*) fornecida por AT&T Laboratories. Alternativamente, as comunicações entre múltiplos clientes X e servidores X podem ser multiplexadas. Essa abordagem é usada na solução XMX desenvolvida por Brown University. Entretanto, ambas abordagens requerem que o aplicativo seja inicializado dentro do servidor X de forma a compartilhar o aplicativo. Isso significa que o usuário precisa reconhecer um desejo para compartilhar ou acessar remotamente um aplicativo antes que ele seja lançado. Essa limitação pode degradar a produtividade, quando um aplicativo não pode ser prontamente reinicializado. Uma outra abordagem para o compartilhamento de aplicativo é adicionar extensões funcionais ao servidor X. Entretanto, o uso de extensões limita severamente o número de plataformas nas quais essa abordagem pode ser prontamente implementada. Categorizar e mapear cada pixel e um outro

problema de desempenho para as abordagens de compartilhamento baseadas em Windows X que buscam compartilhamento dados de exibição, usando funções externas (isto é, sem proxies, apenas protocolos padrão X11), quando as várias janelas dentro da área de exibição compartilhada podem usar diferentes formatos de exibição para representar os dados de exibição. Abordagens externas que tentam endereçar o uso de formatos exibição diferentes falham em fornecer uma solução eficiente para mapear dados de pixel que tomam vantagem da relação de hierarquia das janelas.

[0009] Como um resultado, existe aí uma necessidade de uma maneira para mapear, de forma eficiente, dados de exibição para uma área de exibição na qual múltiplas janelas são apresentadas e mais que um formato de exibição é usado para representar os dados de exibição.

Descrição da Invenção

[0010] A invenção provê um método, sistema e produto de programa de computador para mapeamento de dados de exibição. Uma hierarquia de nós é gerada e representa várias janelas e as respectivas informações de atributo dos mesmos relevantes para a área de exibição que está sendo mapeada. Os dados de exibição para a área de exibição são obtidos e mapeados, de forma eficiente, usando a hierarquia de nós. Os dados de exibição mapeados podem, então, ser usados, por

exemplo, para implementar a colaboração entre usuários em múltiplos sistemas. Por exemplo, um aplicativo Windows X pode ser compartilhado sem a necessidade de modificar o local em que o aplicativo é executado ou o software do servidor X. Além disso, a invenção pode permitir que um sistema compartilhe dados de exibição com diferentes tipos de sistemas (isto é, Windows X e Microsoft Windows®).

[0011] Um primeiro aspecto da invenção provê um método para mapeamento de dados de exibição para uma área de exibição, incluindo pelo menos uma janela, o método compreendendo: criar um nó para cada janela, cada nó incluindo dados de relação hierárquica; gerar uma hierarquia de nós com base nos dados de relação hierárquica, e mapear os dados de exibição a partir de um primeiro formato para um formato compartilhado usando a hierarquia de nós.

[0012] Um segundo aspecto da invenção fornece um sistema para compartilhamento de dados de exibição para uma área de exibição compartilhada incluindo pelo menos uma janela, o sistema compreendendo: meios para criação de um nó para cada janela; meios para gerenciar uma hierarquia de nós que incluem cada nó; meios para mapear os dados de exibição para um formato compartilhado, usando a hierarquia de nós; e meios para compartilhar os dados de exibição mapeados com um computador de destino.

[0013] Um terceiro aspecto da invenção provê um sistema para mapeamento de dados de exibição para uma área de exibição compartilhada, incluindo pelo menos uma janela, o sistema compreendendo: meios para criar um nó para cada janela; meios para gerenciar uma hierarquia de nós que inclui cada nó; e meios para mapear os dados de exibição para um formato compartilhado, usando a hierarquia de nós.

[0014] Um quarto aspecto da invenção provê um produto de programa de computador compreendendo uma mídia utilizável por computador, tendo um código de programa legível por computador embarcado na mesma para mapeamento dos dados de exibição para uma área de exibição incluindo pelo menos uma janela, o produto de programa de computador compreendendo: código de programa configurado para criar um nó para cada janela, o nó incluindo informação de atributo; o código de programa configurado para gerenciar uma hierarquia de nós que inclui cada nó; e código de programa configurado para mapear os dados de exibição para um formato compartilhado, usando a hierarquia de nós.

[0015] Um quinto aspecto da invenção provê uma mídia de armazenamento legível por computador, tendo armazenado na mesma instrução para realizar o método, o método compreendendo as etapas de criar um nó para cada janela, cada nó incluindo dados de relação hierárquica; gerar uma

hierarquia de nós com base nos dados de relação hierárquica, e mapear os dados de exibição a partir de um primeiro formato para um formato compartilhado usando a hierarquia de nós.

[0016] Os aspectos ilustrativos da presente invenção são projetados para resolver os problemas aqui descritos e outros problemas não discutidos, que podem ser descobertos por uma pessoa versada na técnica.

Breve Descrição dos Desenhos

[0017] Essas e outras características dessa invenção serão mais prontamente compreensíveis a partir da descrição detalhada a seguir dos vários aspectos da invenção tomados em conjunto com os desenhos anexos nos quais:

[0018] A figura 1 mostra uma vista de um display compartilhado ilustrativo.

[0019] A figura 2 mostra um diagrama de blocos de um sistema ilustrativo para colaboração entre múltiplos computadores.

[0020] A figura 3 mostra um diagrama de blocos mais detalhado do sistema da figura 2.

[0021] A figura 4 mostra um método de acordo com uma concretização da invenção.

[0022] A figura 5 mostra uma concretização da etapa de mapeamento da figura 4.

[0023] A figura 6 mostra uma representação esquemática de uma hierarquia de nós baseada na exibição compartilhada da figura 1.

[0024] A figura 7 mostra uma representação esquemática alternativa da hierarquia de nós ilustrativa da figura 6.

[0025] A figura 8 mostra uma representação esquemática de uma concretização de uma pluralidade de listas de nós e arrays bidimensionais.

[0026] Deve ser notado que os desenhos da invenção não estão em escala. Os desenhos são pretendidos para representar apenas aspectos típicos da invenção, e, portanto, deve ser considerado como não limitante do escopo da invenção. Nos desenhos, números semelhantes representam elementos análogos entre os desenhos.

Melhor Modo de Realização da Invenção

[0027] A invenção provê um método, sistema e produto de programa de computador para mapeamento de dados de exibição. Uma hierarquia de nós é gerada e representa várias janelas e suas respectivas informações de atributos relevantes para uma área de exibição que está sendo mapeada. Os dados de exibição para a área de exibição são obtidos e mapeados, de forma eficiente, usando a hierarquia de nós. Os dados de exibição mapeados podem, então, ser usados, por

exemplo, para permitir que os dados de exibição sejam compartilhados entre os usuários em múltiplos sistemas como parte de um sistema de colaboração. Enquanto a descrição abaixo descreve a invenção implementada como parte de um sistema de colaboração, é compreendido que o sistema de colaboração é apenas ilustrativo dos vários sistemas nos quais essa invenção pode ser implementada.

[0028] Por questões de conveniência somente, a descrição remanescente inclui três seções denotadas pelos cabeçalhos I. Ambiente, II. Visão Geral do Sistema de Colaboração, e III. Método de Mapeamento.

I. Ambiente

[0029] A figura 1 mostra uma área de exibição ilustrativa 60. A área de exibição 60 pode ser definida usando qualquer método agora conhecido ou a ser desenvolvido subsequentemente. Por exemplo, a área de exibição 60 pode ser definida por meio de um aplicativo, por uma área definida por uma ou mais janelas, por uma área compreendendo uma porção ou toda a área de um display (isto é, a área da tela de um monitor), e/ou por múltiplas áreas disjuntas de um display.

[0030] As janelas 64A-H na área de exibição 60 podem ser relacionadas ou não ter qualquer relação umas com as outras. Por exemplo, cada janela 64A-H pode exibir dados

para um único aplicativo, e, portanto, não ter qualquer relação com qualquer outra janela 64A-H. Alternativamente, algumas ou todas as janelas 64A-H podem ter uma relação hierárquica umas com as outras. Em uma hierarquia de janelas, a janela limitada à área de exibição de uma outra janela é uma "janela filha" da outra "janela pai". Uma hierarquia de janela pode ter qualquer número de níveis, resultando em uma "janela neta" tendo uma janela "avó", tendo uma "janela bisavó" etc. Janelas que compartilham a mesma janela pai ou não possuem janela pai (isto é, são mostradas na área de trabalho) são "janelas irmãs". Deve ser compreendido que as relações entre as janelas 64A-H são discutidas a partir do ponto de vista de um aplicativo. Como um resultado, quaisquer alterações ou modificações na hierarquia que possam ser feitas por um sistema operacional na apresentação da janela não estão endereçadas nessa discussão. Por exemplo, em um sistema Windows X, o gerenciador de janelas pode originar janelas de aplicativos de nível superior, fornecendo decoração ao gerenciador de janelas para permitir à janela movimento, redimensionamento etc.

[0031] Para propósitos de descrição, é considerado que qualquer janela 64A-H que é mostrada completamente dentro de outra janela 64A-H é um filho ou neto daquela janela. Como um resultado, a janela 64B-D são janelas filhas da

janela 64A, as janelas 64E-G são janelas filhas da janela 64B e netas da janela 64A, e a janela 64H é uma janela filha da janela 64D e neta da janela 64A. Consequentemente a janela 64A, as janelas 64B-D, janelas 64E-G, e a janela 64H constituem quatro conjuntos distintos de janelas irmãs. Deve ser entendido que numerosas outras relações entre as janelas 64A-H são possíveis e os ensinamentos da invenção se aplicam igualmente a essas outras relações.

[0032] Uma ordem de empilhamento é atribuída às janelas 64A-H a fim de determinar como as regiões sobrepostas das janelas 64A-H são exibidas. A ordem de empilhamento é parcialmente determinada pela relação hierárquica das janelas. Por exemplo, a janela filha é mostrada no topo de uma janela pai. Consequentemente, as janelas 64B-D são, cada uma, mostradas no topo de suas janelas pai 64A. Dentro de um conjunto de janelas irmãs, a ordem de empilhamento é ainda determinada pela ordem que as janelas foram criadas, selecionadas pelo usuário, atualizadas etc. Conforme mostrado, dentro do conjunto de janelas irmãs 64B-D, a janela 64B é exibida no topo da janela 64C que é exibida no topo da janela 64D. Consequentemente, a janela 64B está no topo da ordem de empilhamento para o conjunto de janelas irmãs, seguido pela janela 64C, e, então, pela janela 64D. Similarmente, dentro do conjunto de janelas irmãs 64E-G, a

janela 64E está no topo da ordem de empilhamento, seguida pela janela 64F e, a seguir, pela janela 64G.

II. Visão Geral do Sistema de Colaboração

[0033] Como discutido anteriormente, o mapeamento de dados de exibição é comumente implementado como parte de um sistema de colaboração de modo que os dados de exibição podem ser compartilhados. A figura 2 mostra um diagrama de blocos de um sistema ilustrativo 10 para colaboração usando múltiplos computadores 12, 16. O sistema 10 inclui um computador fonte 12, tendo um display fonte 14 e um computador de destino 16 tendo um display de destino 18. Deve ser entendido que enquanto apenas dois computadores 12, 16 são mostrados e discutidos, os ensinamentos da invenção se aplicam a sistemas de colaboração implementados usando qualquer número de computadores. Tipicamente, o usuário 20 interage com o computador fonte 12 usando um ou mais dispositivos de E/S que incluem o display fonte 14. O computador fonte 12 pode incluir um sistema de exibição baseado em janela que exibe dados para um ou mais aplicativos nas janelas 22A-D no display fonte 14. Ademais, o computador fonte 12 e/ou o computador de destino 16 inclui um ou mais sistemas para compartilhar dados de exibição para o display fonte 14 com o display de destino 18.

[0034] O display fonte 14 inclui uma área de exibição 23. Uma área de exibição compartilhada 25 pode ser definida dentro da área de exibição 23, usando-se qualquer método, incluindo, por exemplo, por meio de um aplicativo em execução no computador fonte 12, uma ou mais janelas 22A-D exibidas no display fonte 14, e/ou uma área compreendendo uma porção de ou toda a área de exibição 25 do display fonte 14. Nesse exemplo, a área de exibição compartilhada 25 é definida como a porção da área de exibição 23 tendo janelas 22A-C. A área de exibição compartilhada 25 é compartilhada com o computador de destino 16 para exibição no display de destino 18, como área de exibição de destino 125. A janela 22D não é compartilhada com o computador de destino 16. O usuário 24 pode visualizar a área de exibição de destino 125 no display de destino 18 como as janelas 122A-C. O usuário 24 também pode ser permitido interagir com o computador de destino 16 a fim de alterar os dados de exibição para a área de exibição compartilhada 25, que é subsequentemente mostrada na área de exibição de destino 125.

[0035] A figura 3 mostra um diagrama de blocos mais detalhado do sistema 10. O computador fonte 12 é mostrado, incluindo uma unidade de processamento central (CPU) 30, memória 32, interface de entrada/saída (E/S) 34, barramento 36, e uma base de dados opcional 38. O computador de destino

16 é mostrado em comunicação com o computador fonte 12. A comunicação entre o computador fonte 12 e o computador de destino 16 pode ser qualquer mecanismo conhecido ou a ser desenvolvido subsequentemente para tais propósitos, por exemplo, uma ou mais conexões de fio rígido direta (por exemplo, porta serial), ou através de uma conexão endereçável em um ambiente cliente-servidor (ou servidor-servidor), que pode utilizar qualquer combinação de métodos de transmissão por cabo e/ou sem fio. Em um ambiente cliente-servidor, o servidor e o cliente podem ser conectados através da Internet, uma rede de área ampla (WAN), uma rede de área local (LAN), uma rede privada virtual (VPN) ou outra rede privada. O servidor e o cliente podem utilizar conectividade de rede convencional, tal como Token Ring, Ethernet, WiFi ou outros padrões de comunicação convencionais. Onde o cliente se comunica com o servidor através da Internet, a conectividade pode ser proporcionada por protocolos baseados em sockets de TCP/IP convencionais. Nessa instancia, o cliente poderia utilizar um serviço de provedor de Internet para estabelecer conectividade com o servidor.

[0036] O computador fonte 12 e o computador de destino 16 podem compreender qualquer sistema de propósito geral ou uso específico que utiliza software de sistema operacional padrão, que é projetado para acionar a operação

do hardware particular e que é compatível com os outros componentes do sistema e controladores de E/S. A CPU 30 pode compreender uma única unidade processamento, múltiplas unidades de processamento capazes de operação paralela ou ser distribuída através de uma ou mais unidades de processamento em uma ou mais localizações, por exemplo, em um cliente e servidor. A memória 32 pode compreender qualquer tipo conhecido de mídia de armazenamento de dados e/ou transmissão, incluindo mídias magnéticas, mídias óticas, memórias de acesso aleatório (RAM) e memórias somente de leitura (ROM), uma cache de dados, um objeto de dados etc. Além disso, similarmente a CPU 30, a memória 32 pode residir em uma localização física, compreendendo um ou mais tipos de armazenamento de dados, ou ser distribuída através de uma pluralidade de sistemas físicos em várias formas.

[0037] Conforme mostrado, o usuário 20, o display fonte 14 e o computador de destino 16 interagem com o computador fonte 12 através da interface de E/S 34. A interface de E/S 34 pode compreender qualquer sistema para troca de informações com o usuário 20, display fonte 14, e/ou computador de destino 16, incluindo, por exemplo, uma porta de E/S (serial, paralela, ethernet, teclado, mouse etc.), uma porta de barramento serial universal (USB), barramento de expansão, dispositivos eletrônicos de

acionamento integrado (IDE), um sistema de rede, um modem, um alto-falante, um monitor (tubo de raio catódico (CRT), display de cristal líquido (LCD) etc.), dispositivo *hand-held*, teclado, mouse, sistema de reconhecimento de voz, sistema de saída de expressão, scanner, impressora, fax, pager, dispositivos de armazenamento etc. o barramento 36 fornece uma ligação de comunicação entre cada um dos componentes no computador fonte 12 e, de forma semelhante, pode compreender qualquer tipo conhecido de ligação de transmissão, incluindo elétrica, ótica, sem fio etc. Ademais, embora não mostrados, componentes adicionais, tal como memória cache, sistemas de comunicação, software de sistema etc. podem ser incorporadas em computador fonte 12.

[0038] A base de dados 38 pode fornecer armazenamento para informação necessária a realização da invenção como aqui descrito. Como tal, a base de dados 38 pode incluir um ou mais dispositivos de armazenamento, tal como um drive magnético ou ótico. Além disso, a base de dados 38 pode incluir dados distribuídos ao longo, por exemplo, de uma LAN, WAN ou rede de área de armazenamento (SAN) (não mostrada). A base de dados 38 pode também ser configurada de tal maneira que alguém habilitado na técnica pode interpretá-la para incluir um ou mais dispositivos de armazenamento. Deve ser entendido que o computador de destino 16 também

inclui os vários componentes de hardware mostrados e discutidos em conjunto com o computador fonte 12. Esses componentes não foram representados ou discutidos separadamente por questões de brevidade.

[0039] A fim de implementar as várias funções do sistema de colaboração 10, o computador fonte 12 inclui programa de colaboração 40A armazenado na memória 32 como código de programa de computador, e o computador de destino 16 inclui programa de colaboração 40B armazenado como código de programa de computador. O programa de colaboração 40A inclui um sistema de mapeamento 41 e um sistema de compartilhamento 54. Como será descrito ainda abaixo, o sistema de mapeamento 41 mapeia os dados de exibição para área de exibição compartilhada 25 (figura 2) a partir de um primeiro formato em um formato compartilhado. O sistema de compartilhamento 54 armazena os dados de exibição mapeados, como dados de exibição compartilhados que são subsequentemente fornecidos para o programa de colaboração 40B no computador de destino 16. O sistema de mapeamento 41 pode atualizar os dados de exibição compartilhados e o sistema de compartilhamento 54 pode compartilhar os dados de exibição compartilhados periodicamente, sob demanda, ou alguma combinação dos mesmos.

[0040] O programa de colaboração 40B inclui sistema de exibição 56 que é configurado para exibir os dados de exibição compartilhados na área de exibição de destino 125 (figura 2) no display de destino 18. Quando os dados de exibição compartilhados, o sistema de exibição 56 pode determinar se alguns dos dados de exibição compartilhados foram modificados, e atualizar as porções da área de exibição de destino 125 que foram modificadas. A determinação pode ser feita, por exemplo, por meio da divisão dos dados de exibição em seções e comparar uma seção de cada vez. O sistema de compartilhamento 54 pode monitorar a frequência que os dados de exibição compartilhados estão sendo usados pelo computador de destino 16 e ajustar a taxa na qual o sistema de mapeamento 41 mapeia os dados de exibição com base na frequência monitorada. Nesse caso, o período de tempo máximo pode ser estabelecido além do que os dados de exibição compartilhados são atualizados com os dados de exibição recentemente mapeados. O programa de colaboração 40B também inclui sistema remoto 58 que permite que o usuário 24 altere os dados de exibição para a área de exibição compartilhada 25, usando o computador de destino 16. A fim de implementar a alteração feita pelo usuário, o sistema remoto 58 coleta eventos de dispositivos de entrada (por exemplo, teclado, mouse etc.) gerados no computador de destino 16 e encaminha

esses eventos para o sistema de compartilhamento 54 no computador fonte 12. O sistema de compartilhamento 54 posiciona os eventos no fluxo de eventos do computador fonte 12 de forma que eles são processados como se eles fossem gerados por meio de um dispositivo de entrada localizado no computador fonte 12.

[0041] O sistema de mapeamento 41 mapeia os dados de exibição para a área de exibição compartilhada 25 (figura 2) em dados de exibição compartilhados usados pelo sistema de compartilhamento 54. Em particular, o sistema de mapeamento 41 mapeia os dados de pixel incluídos nos dados de exibição para cada pixel na área de exibição 25 a partir de um primeiro formato em um formato compartilhado. O sistema de mapeamento 41 inclui um sistema de definição 42, um sistema de nó 44, um sistema de gerenciador 46, um sistema de monitor 48 e um sistema de recuperação 50, e um sistema de conversão 52. A operação desses sistemas será descrita abaixo em relação ao método de mapeamento da invenção.

[0042] Deve ser entendido que menos/adicionais funcionalidades podem ser incluídas no sistema de colaboração 10. Por exemplo, o sistema remoto 58 permite que o usuário 24 altere os dados de exibição para área de exibição compartilhada 25, usando o computador de destino 16, uma característica opcional do sistema de colaboração

10. Ademais, o usuário 20 e/ou usuário 24 pode ser permitido selecionar a extensão da colaboração implementada entre os sistemas. Uma sequência de toques de teclas pode ser usada para especificar que nenhuma colaboração, display compartilhado apenas, display compartilhado e áudio, controle remoto etc. deve ocorrer. Ademais, a localização de um apontador de mouse em um display (display fonte 14 e/ou display de destino 18) pode ser compartilhada entre o computador fonte 12 e o computador de destino 16, usando o sistema de compartilhamento 54 e/ou o sistema remoto 58.

III - Método de mapeamento

[0043] A invenção mapeia dados de exibição para uma área de exibição em um formato compartilhado. Em uma concretização, os dados de exibição incluem dados de pixel para cada pixel na área de exibição. Dados de pixel incluem informação sobre a localização do pixel, cor do pixel, e a profundidade do pixel. Os dados de pixel são mapeados a partir de um primeiro formato para um formato compartilhado. O método da invenção é discutido com referência às figuras 3 e 4, que, respectivamente, mostram um sistema de mapeamento ilustrativo 41 e uma visão geral das etapas do método usado na realização do mapeamento de acordo com uma concretização da invenção. Na etapa S1 (figura 4), uma área de exibição 25

(figura 2) a ser mapeada é definida pelo sistema de definição 42 (figura 3).

[0044] Na etapa S2 (figura 4), um nó é criado para cada janela que é relevante para a área de exibição 25 pelo sistema de nó (figura 3). Em uma concretização, um nó é criado para cada janela mostrada no display fonte 14 (isto é, janelas 22A-D na figura 2). A inclusão de todas as janelas permite que a localização de uma janela que não é compartilhada (isto é, janela 22D) seja monitorada a fim de determinar se ela é subsequentemente movida para uma área compartilhada ou sobrepõe uma janela compartilhada. Entretanto, deve ser entendido que os nós podem ser criados apenas para uma porção da janela (por exemplo, todas as janelas dentro de uma certa área do display fonte 14 ou todas as janelas designadas como sendo compartilhadas). A figura 6 apresenta uma representação esquemática de uma hierarquia de nós 62 para a área de exibição 60 da figura 1 que inclui nós 66A-H. Cada nó 66A-H é uma coleção de dados que inclui informação de atributo para sua janela correspondente 64A-H (figura 1). A "informação de atributo" inclui dados de relação hierárquica (por exemplo, dados sobre uma janela pai e/ou janela (s) filha (s)), bem como o tamanho, localização e/ou forma da janela correspondente. Além disso, dados para o formato dos dados de pixel para a janela são armazenados

como informação de atributo. Em uma concretização ilustrativa, os dados de formato incluem uma profundidade e visual para os dados de pixel, conforme mostrado na figura 6. Ademais, um mapa de cor pode ser criado e armazenado como informação de atributo para um nó que corresponde a janela usando valores de pixel indexados. Frequentemente, as janelas em um aplicativo compartilham um único mapa de cores. Consequentemente, um mapa de cor pode ser criado e compartilhado por todos os nós correspondendo a janelas que compartilham o mapa de cor. Para tipos de visual dinâmico, o mapa de cor pode precisar ser atualizado com base na ocorrência de um evento de janela que indica que a instância de mapa de cor foi alterada e/ou em uma base periódica para determinar se algum valor no mapa de cor foi modificado. Por exemplo, para um mapa de cor dinâmico, os valores no mapa de cor podem ser lidos novamente toda vez que a área de exibição é mapeada novamente. Deve ser entendido que mais ou menos informação pode ser armazenada como informação de atributo. Por exemplo, dados de ordem de empilhamento, informação de informação de recorte (por exemplo, a porção de uma janela visível dentro de sua transparência) e/ou tamanho da borda podem ser incluídos para cada janela.

[0045] Na etapa S3 da figura 4, uma hierarquia de nós 62 (figura 6) é gerada, usando-se dados de relação

hierárquica e/ou dados de ordem de empilhamento como parte da informação de atributo de cada nó 66A-H pelo sistema gerenciado 46 (figura 3). A figura 7 mostra uma concretização ilustrativa da hierarquia de nós 62, em que cada nó 66A-H inclui informação de atributo para a janela correspondente 64A-H (figura 1), bem como um ponteiro para um nó pai, um nó irmão e um nó filho para implementar uma estrutura hierárquica. O uso de ponteiros permite navegação eficiente e gerenciamento de hierarquia de nós 62, incluindo reordenação, inserção e exclusão de nós, como é bem conhecido na técnica. O uso de e configuração de ponteiros para gerenciamento de dados é bem conhecido na técnica e, portanto, não será mais discutido aqui. Deve estar entendido que ponteiros adicionais ou menos ponteiros podem ser usados, bem como qualquer método alternativo de gerenciamento de dados com ou sem ponteiros.

[0046] Os nós 66A-H são inicialmente posicionados na hierarquia de nós 62 com base nos dados de relação hierárquica. Fazendo referência as figuras 1, 6 e 7 juntas, uma vez que a janela 64A não tem uma janela pai, seu nó correspondente, nó correspondente 66A, é posicionado no topo da hierarquia de nós 62. Os nós 66B-D para janelas filhas 64B-D da janela 64A são posicionados um nível abaixo do nó 66A, os nós 66E-G para janelas filha 64E-G da janela 64B são

posicionados um nível abaixo do nó 66B e o nó 66EH para janela filha 64H da janela 64D é posicionado um nível abaixo do nó 66D. Nessa concretização, um conjunto de nós irmão (isto é, o conjunto de nós para um conjunto de janelas irmãs) é ainda posicionado na hierarquia de nós 62, de acordo com os dados de ordem de empilhamento para o conjunto de janelas irmãs. O conjunto de janelas irmãs é configurado de modo que o nó mais alto na ordem de empilhamento é posicionado primeiro com os nós irmãos remanescentes seguindo a ordem decrescente. Consequentemente, para as janelas 64B-D, o nó 66B é posicionado primeiro, seguido pelo nó 66C, e, então pelo nó 66D. Similarmente, para a janela 64E-G, o nó 66E é posicionado primeiro, seguido pelo nó 66F, e, então pelo nó 66G.

[0047] Uma vez criada, a hierarquia de nós 62 e a informação de atributo em nós individuais 66A-H pode prontamente se tornar desatualizada. Consequentemente, as etapas 84 e 85 estão incluídas na figura 4. Na etapa S4, vários tipos de eventos de janela são monitorados pelo sistema monitor 48 (figura 3). Na etapa S5, a hierarquia de nós é atualizada com base na ocorrência de um dos tipos de evento de janela monitorados pelo sistema monitor 46 (figura 3). Por exemplo, fazendo referência à figura 1, um usuário pode selecionar a janela 64H, movendo, desse modo, sua janela

pai 64D para frente da ordem de empilhamento dentro do conjunto de irmãos 64B-D. Além disso, uma ou mais janelas pode ser criada, destruída, movida, redimensionada etc. Em geral, sistemas de exibição de janelas geram eventos de janela quando uma ou mais ações ocorrem. Consequentemente, a ocorrência desses eventos pode ser monitorada, de forma assíncrona, e a hierarquia de nós 62 e/ou a informação de atributo em um ou mais nós 66A-H pode ser atualizada com base na ocorrência de um evento de janela monitorado.

[0048] As etapas S6 e S7 da figura 4 são mostradas ocorrendo em paralelo com as etapas S4 e S5. A etapa S6, dados de exibição são obtidos pelo sistema de recuperação (figura 2). Na etapa S7, os dados de exibição são mapeados em um formato compartilhado, usando a hierarquia de nós pelo sistema de conversão 52 (figura 2). Cada conjunto de etapas S4-S5 e S6-S7 é repetido até que é determinado nas etapas S8 e S9, respectivamente que o mapeamento dos dados de exibição não é mais desejado. Enquanto mostrado em paralelo, deve ser entendido que as etapas S4-S5 e S6-S7 podem ser implementadas em paralelo, série ou alguma combinação.

[0049] Com referência à etapa S6 e figuras 1, 6 e 7 coletivamente, o uso de hierarquia de nós 62 permite que dados de exibição obtidos para uma janela antecessora sejam usados no mapeamento de uma ou mais janelas filhas, janelas

netas, etc. Para obter dados de exibição, os dados de exibição para a área de exibição definida pela janela de nível superior (isto é, a janela 64A na figura 1) são copiados de uma memória de exibição. Uma vez que cada janela é localizada dentro da área de exibição de sua janela pai, os dados de exibição para algumas ou todas as janelas filha, janelas netas, etc. (isto é, janelas 64B-H) podem ser também obtidos.

[0050] Quando implementado em um sistema X Windows, os dados de exibição para o nó(s) que compartilham uma profundidade comum com um nó antecessor e cada nó antecessor interveniente pode ser obtido em uma única operação que adquire os dados de exibição para o nó antecessor. Por exemplo, a figura 6 mostra nós 66A, 66B, 66E, tendo oito dados de pixel de bit de pseudocor (PC - PseudoColor), os nós 66C, 66H, tendo vinte e quatro dados de pixel de bit de PC, os nos 66D, 66G, tendo vinte e quatro dados de pixel de bit de cor verdadeira (TrueColor - TC) e o nó 66F, tendo oito dados de pixel de bit de PC. Desde que a profundidade dos dados de pixel seja a mesma, os dados de exibição para múltiplos nós podem ser obtidos em uma operação, independente do visual que está sendo usado. Como um resultado, quando dados de exibição para o nó 66A são obtidos, os dados de exibição para os nós 66B, 66E e 66F são também obtidos. Os

dados de exibição para o nó 66F podem ser obtidos ainda que o nó 66F use dados de pixel de TC, enquanto seus nós antecessores 66A, 66B usam dados de pixel de PC.

[0051] No entanto, em um sistema X Windows, quando uma janela ou janela antecessora interveniente usa uma profundidade diferente que a da janela antecessora, os dados de exibição para a janela são indefinidos, quando os dados de exibição para sua janela antecessora são recuperados. Nesse caso, múltiplas operações são necessárias para obter os dados de exibição para todos os nós 66A-H. Por exemplo, os nós 66G usam dados de pixel que têm vinte e quatro bits. Consequentemente, uma operação separada é necessária para obter seus dados de pixel. De maneira similar, os nós 66C, 66D também usam dados de pixel que têm vinte e quatro bits. Cada nó 66C, 66D requer uma operação adicional para obter seus dados de exibição, uma vez que nenhum nó 66C, 66D é um antecessor do outro. No entanto, quando os dados de exibição para o nó 66D são obtidos, os dados de exibição para o nó 66H é também obtido, uma vez que ele usa a mesma profundidade.

[0052] Uma vez que os dados de exibição tenham sido obtidos, a etapa S7 (figura 4) mapeia os dados de pixel dentro dos dados de exibição de seu formato corrente para um formato compartilhado. A figura 5 mostra uma concretização

ilustrativa da etapa de mapeamento, em que a etapa S7A determina o nó correspondente para cada pixel, a etapa S7B determina o formato corrente dos dados de pixel para o pixel que usa o nó e a etapa S7C gera dados de pixel em um formato compartilhado dos dados de pixel em seu formato corrente. A etapa S7D repete essas etapas para cada pixel nos dados de exibição. A geração da hierarquia de nós 67 (figuras 6 e 7) com nós 66A-H e o armazenamento de informação de atributo para a janela correspondente 66A-H em cada nó 66A-H permite que o nó correto seja localizado, de forma eficiente, na etapa S7A. Para localizar o nó apropriado, a hierarquia de nós 62 é transversa em uma ordem prefixada. Ou seja, o primeiro nó que contém um pixel e para o qual o pixel não está dentro de qualquer de seus nós filhos, netos etc., é o nó correto para uso no mapeamento de dados de pixel para o pixel.

[0053] O pixel 68 na figura 1 proporciona um exemplo ilustrativo das etapas S6 e S7 (figura 4). O pixel 68 está dentro da área definida por cada janela 64A-H, entretanto, a janela 64E detém correntemente o pixel 68. Portanto, o mapeamento de dados de pixel para o pixel 68 de um formato para o formato compartilhado requer o uso de informação de atributo no nó 66E. A fim de determinar que o nó 66E corresponde ao pixel 68, o nó no nível de topo da hierarquia

é inicialmente consultado. A informação de atributo no nó 66A é usada para determinar se o pixel 68 está dentro da área da janela 64A. Uma vez que ele esteja, o nó 66B, o primeiro filho do nó 66A, é consultado a seguir. Uma vez que o pixel 68 está dentro da área da janela 64B, o nó 66E, o primeiro filho do nó 66B, é consultado. É, então, determinado que o pixel 68 está dentro da área da janela 64E e o nó 66E não tem filhos. Portanto, o nó 66E é o nó que está associado com o pixel 68.

[0054] Agora que dados de pixel e seus nós correspondentes estão disponíveis, o formato corrente dos dados de pixel é determinado, usando-se a informação de atributo do nó (etapa S7B da figura 5) e os dados de pixel no formato compartilhado são gerados (etapa S7C da figura 5). Em uma concretização, a profundidade, o visual e/ou informação de mapa de cor no nó são usados para mapear os dados de pixel em um formato compartilhado. O formato compartilhado pode compreender, por exemplo, um formato de vinte e quatro bits de cor verdadeira que pode ser prontamente traduzido por outros aplicativos e/ou sistemas. Nesse caso, quando os dados de pixel incluem os valores de RGB nos próprios dados, o mapeamento de dados pode compreender o deslocamento de dados de pixel de modo que cada valor de RGB seja representado por oito bits. Por dados

de pixel que usa indexação, o pixel pode ser usado para procurar os valores de RGB no mapa de cor, e os valores de RGB são subsequentemente deslocados de modo que cada valor tem oito bits.

[0055] Para aumentar ainda mais a velocidade com que um nó apropriado é localizado, a área de exibição pode ser dividida em uma pluralidade de seções. Por exemplo, a figura 1 mostra a área de exibição 60 dividida em duas grades bidimensionais com cada seção 70A-C sendo um conjunto de números de pixel amplo e longo. Em uma concretização, cada seção 70A-C é um bloco quadrado de pixels tendo um tamanho que é uma potência de dois, por exemplo, sessenta e quatro bits amplos por sessenta e quatro bits longos. A cada Seção 70A-C pode ser atribuído um endereço (isto é, coordenadas x e y) similar aos pixels na tela de exibição. O uso de uma largura e profundidade que são potências de dois permite que o endereço da seção em que um pixel localizado por ser prontamente determinando por deslocamento à direita dos valores que representam localização do pixel. Deve ser entendido, no entanto, que qualquer tamanho ou formato de seções 70A-C pode ser usado.

[0056] Fazendo referência as figuras 1 e 6-8 juntas, usando a hierarquia de nós 62 e seções 70A-C uma lista de nós 76A-C pode ser criada para cada seção 70A-C. Cada lista

de nós 76A-C inclui o nó para cada janela correntemente visível dentro da seção 70A-C. Em uma concretização, um arranjo bidimensional 74 é criado em que cada elemento do arranjo 72A-C é associado com uma única seção 70A-C. Cada elemento do arranjo 72A-C tem uma lista de nós 76A-C para sua seção 70A-C. Cada lista de nó 76A-C inclui cada nó que detém pelo menos um pixel dentro da seção 70A-C para o elemento 72A-C. Por exemplo, a lista de nós 76A para o elemento 72A inclui cada nó que é visível dentro da seção 70A. Cada lista de nós 76A-C é ordenada tal que o nó que corresponde à janela mais alta dentro de sua seção 70A-C está localizada no início da lista de nós 76A-C com os nós remanescentes localizados em ordem decrescente. Alternativamente, um nó que corresponde à janela que correntemente detém mais pixels dentro de sua seção 70A-C pode ser localizado no início de cada lista de nós 76A-C.

[0057] Enquanto manutenção de dados adicional é requerida para listas de nós 76A-C, elas reduzem o número médio de nós que devem ser buscados para determinar o nó apropriado para um dado pixel. Por exemplo, a seção 70B é uma seção de área de exibição 60. Janelas 64B, 64C, 64E e 64G detém pelo menos uma porção da área de exibição dentro da seção 70. Consequentemente, o elemento do array 72B tem a lista de nós 76B que inclui nós 66B, 66C, 66E e 66G. Como

discutido acima, a lista de nós 76B é ordenada de modo que o nó no topo da ordem de empilhamento está localizado primeiro. Nesse caso, o nó 66E é o primeiro nó, seguido pelo nó 66G, o nó 66b, e, a seguir, o nó 66C. A fim de localizar o nó que corresponde ao pixel 68, é determinado que o pixel 68 está localizado dentro da seção 70B. Usando-se a lista de nós 76B para o elemento 72B, é determinado que o nó 66E é o nó apropriado após consultar apenas um único nó.

[0058] O uso de seções 70A-C e lista de nós 76A-C permite prontamente que dados de exibição sejam seções obtidas por seção ao invés de por janela. Quando a seção é obtida por seção, a informação de atributo de cada nó 66A-H pode incluir ainda dados que identificam um. O nó antecedente mais alto corresponde à janela antecessora mais adiante, para a qual os dados de exibição para a janela que correspondem ao nó podem ser obtidos, usando-se uma única operação. Conforme discutido acima, no X Windows, quando dados de exibição para uma janela são armazenados usando-se uma profundidade diferente que sua janela pai, duas operações são necessárias para obter os dados de exibição para cada janela. Como um resultado, o nó 66A seria o nó antecessor mais alto para os nós 66A, 66B, 66E, 66F, nós 66C, 66d, 66G teriam eles mesmos como seus nós antecessores mais altos e o nó 66H teria o nó 66D como o seu nó antecessor mais alto.

A inclusão do nó antecessor mais alto permite que o benefício da hierarquia de nós 62 seja prontamente explorado em uma base seção por seção. Por exemplo, quando da obtenção de dados de exibição para seção 70A, a lista de nós 76A é consultada. O nó 66E é o primeiro nó e tem um antecessor mais alto do nó 66A. Consequentemente, os dados de exibição são obtidos para a porção da janela 64A que está dentro da seção 70A. Os dados de exibição obtidos são usados para mapear os dados de exibição para as janelas 64E, 64F e 64B. Uma segunda operação é requerida para recuperar os dados de exibição para a janela 64C, e uma terceira operação é requerida para recuperar dados de exibição para janela 64H, uma vez que essas janelas usam uma profundidade diferente da janela 64A.

[0059] A eficiência de mapeamento de dados de exibição pode ser ainda aumentada pela inclusão de um segundo array bidimensional 78A para cada elemento de array 72A-C. O array 78A inclui um elemento de array 80 para cada pixel localizado dentro da seção correspondente 70A. Cada elemento de array 80 inclui um identificador que identifica o nó que corresponde à janela que detém o pixel. Quando uma janela que não é compartilhada é exibida dentro da seção 70A-C, os elementos 80 no array 78A que correspondem aos pixels detidos pela janela podem ser marcados como inválidos. O uso de

arrays 78A assegura que o nó correspondente para qualquer pixel na área de exibição compartilhada pode ser encontrado em uma única operação.

[0060] As listas de nós 76A-C e/ ou arrays 78A para seções 70A-C podem ser atualizadas junto com a hierarquia de nós 62, quando da ocorrência de um evento de janela monitorado é detectado que efetua uma ou mais listas de nós 76A-C e/ ou arrays 78A. Em uma concretização, os eventos de janelas monitorados são enfileirados até que a atualização de dados de exibição compartilhados está para recomeçar, na hora em que a hierarquia de nós 62, a lista de nós 76A-C e os arrays 78A são atualizados com base em todos os eventos de janela enfileirados, se necessário. As listas de nós 76A-C e arrays 78A podem ser implementados usando-se qualquer método agora conhecido ou a ser desenvolvido. Por exemplo, cada lista de nós 76A-C pode compreender uma lista de ponteiros encadeados para o nó correspondente armazenado na memória e cada array 78A pode compreender um array bidimensional tendo elementos que contém um ponteiro para o nó correspondente armazenado na memória para cada pixel, como é bem conhecido na técnica.

[0061] Enquanto a discussão está limitada a nós criados para cada janela na área de exibição a ser mapeada, deve ser entendido que um nó pode ser criado e gerenciado

para todas as janelas em uma tela de exibição, independentemente se eles estão localizados dentro da área de exibição a ser mapeada. Além disso, enquanto o método é descrito com referência aos dados de pixel, deve ser entendido que os ensinamentos da invenção se aplicam a qualquer tipo de dados de exibição. Ainda mais, deve ser entendido que um nó pode ser criado para a área de trabalho (ou seja, o plano de fundo) da tela de exibição para obter dados de visualização para porções da tela de exibição não cobertas por qualquer janela.

Aplicabilidade Industrial

[0062] A invenção é aplicável a sistemas de servidor/computador de propósito geral, onde o mapeamento de dados de exibição seja desejável. Deve ser apreciado que a invenção pode ser feita em hardware, software ou uma combinação de hardware e software. Uma combinação típica de hardware e software poderia ser um sistema de computador de propósito geral com um programa de computador que, quando carregado e executado, controla o computador fonte 12 (figura 2) e/ou computador de destino 16, tal que eles realizam os respectivos métodos aqui descritos. Alternativamente, um computador de uso específico, contendo hardware especializado para realizar uma ou mais tarefas funcionais da invenção, poderia ser utilizado.

[0063] A invenção pode também ser embarcada em um produto de programa de computador que compreende todas as respectivas características que permitem a implementação dos métodos descritos aqui e que - quando carregado em um sistema de computador - é capaz de realizar esses métodos. O programa de computador, programa de software, programa ou software, no presente contexto significam qualquer expressão em qualquer linguagem, código ou notação de um conjunto de instruções pretendido para fazer com que um sistema tendo uma capacidade de processamento de informação realize a função particular diretamente ou após um ou ambos o seguinte:

- a) conversão para uma outra linguagem, código ou notação; e
- b) reprodução em uma forma material diferente.

[0064] Enquanto vários métodos foram descritos como ocorrendo em uma sequência particular, deve ser entendido que as etapas independentes podem ser feitas simultaneamente ou em uma ordem diferente da que está aqui descrita. Além disso, vários sistemas foram esquematizados e discutidos como implementação de várias funções, deve ser entendido que menos e/ou sistemas adicionais podem ser usados para implementar as várias funções aqui descritas.

[0065] A descrição precedente de vários aspectos da invenção foi apresentada com propósito de ilustração e descrição. Não é pretendido que esta seja exaustiva ou limite

a invenção à forma precisa apresentada, e, obviamente, muitas modificações e variações são possíveis. Tais modificações e variações que podem ser aparentes para uma pessoa versada na técnica são pretendidas para estar incluídas dentro do escopo da invenção, conforme definido pelas reivindicações anexas.

REIVINDICAÇÕES

1) Método para mapeamento de dados de exibição para uma área de exibição (60) incluindo pelo menos uma janela (60A-H), o método compreendendo:

criar (S2) um nó (66A-H) para cada janela, cada nó incluindo dados de relação hierárquica;

gerar (S3) uma hierarquia de nós (62) com base nos dados de relação hierárquica; e

mapear (S7) os dados de exibição a partir de um primeiro formato obtido pelo uso de atributo do nó para um formato compartilhado que pode ser traduzido por outros aplicativos usando a hierarquia de nós, os dados de exibição incluindo dados de pixel incluindo informação sobre pelo menos um de localização de pixel, cor de pixel, profundidade de pixel, o método **caracterizado pelo** fato de que cada um dos nós é associado com pelo menos uma seção da área de exibição.

2) Método, de acordo com a reivindicação 1, **caracterizado pelo** fato de ainda compreender:

monitorar (S4) uma ocorrência de um evento de janela; e

atualizar (S5) a hierarquia de nós (62) com base na ocorrência de evento de janela.

3) Método, de acordo com a reivindicação 2, **caracterizado pelo** fato de ainda compreender a atualização

da relação hierárquica com base na ocorrência de um evento de janela.

4) Método, de acordo com a reivindicação 1, **caracterizado pelo** fato de que o conjunto de nós irmãos (66B-D) para um conjunto de janelas irmãs (64B-D) está localizado na hierarquia de nós (62) com base em uma ordem de empilhamento das janelas irmãs.

5) Método, de acordo com a reivindicação 1, **caracterizado pelo** fato de ainda compreender:

dividir a área de exibição (60) em uma pluralidade de seções (70A-C); e

criar uma lista de nós (76A-C) para cada seção usando a hierarquia de nós (62), em que o nó para cada janela exibida em pelo menos uma porção da seção é incluído na lista de nós.

6) Método, de acordo com a reivindicação 5, **caracterizado pelo** fato de ainda compreender a criação de um array (74) tendo um elemento (72A-C) para cada seção, em que cada elemento inclui a lista de nós para a seção correspondente.

7) Método, de acordo com a reivindicação 5, **caracterizado pelo** fato de ainda compreender:

monitorar (S4) uma ocorrência de um evento de janela;
e

atualizar (S5) pelo menos uma das listas de nós com base na ocorrência de um evento de janela.

8) Método, de acordo com a reivindicação 1, **caracterizado pelo** fato de ainda compreender a obtenção (S6) de dados de exibição para uma pluralidade de janelas tendo uma primeira profundidade de pixel em uma operação única.

9) Método, de acordo com a reivindicação 8, **caracterizado pelo** fato de ainda compreender a repetição das etapas de obtenção e mapeamento (S6, S7) para pelo menos uma janela tendo uma segunda profundidade de pixel.

10) Método, de acordo com a reivindicação 1, **caracterizado pelo** fato de a etapa de mapeamento (87) incluir:

determinar (S7A) o nó correspondente para cada pixel nos dados de exibição;

determinar (S7B) um primeiro formato de dados de pixel para cada pixel usando o nó; e

gerar (S7C) dados de pixel no formato compartilhado para cada pixel a partir dos dados de pixel no primeiro formato.

11) Método, de acordo com a reivindicação 1, **caracterizado pelo** fato de ainda compreender o compartilhamento de dados de exibição compartilhados com um computador de destino (16).

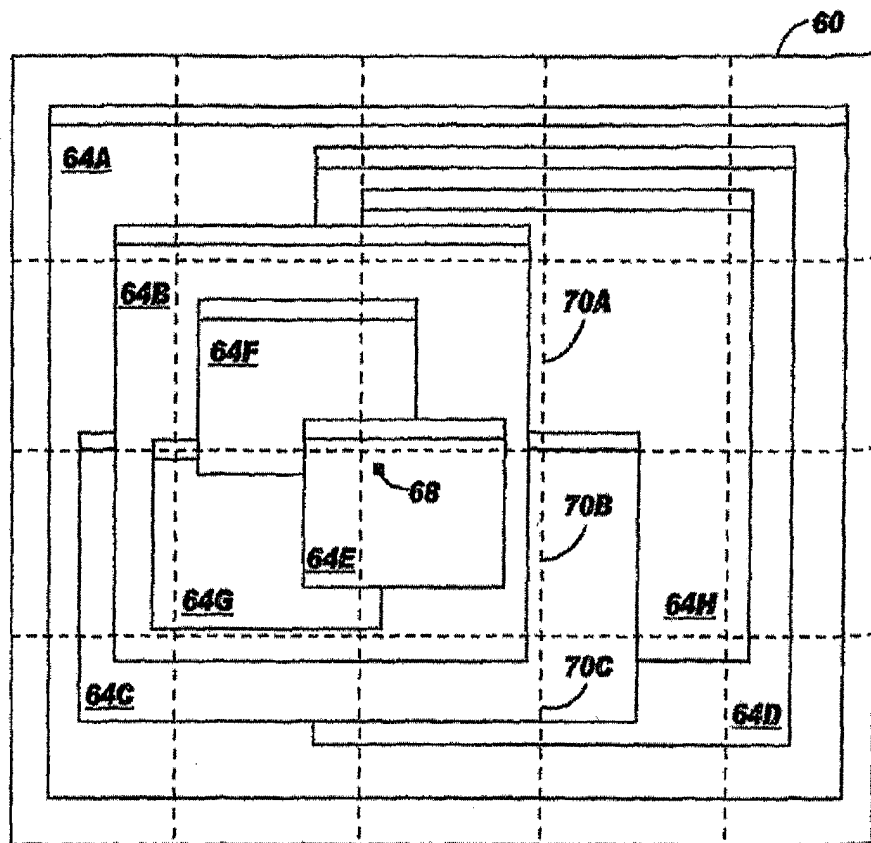


FIGURE 1

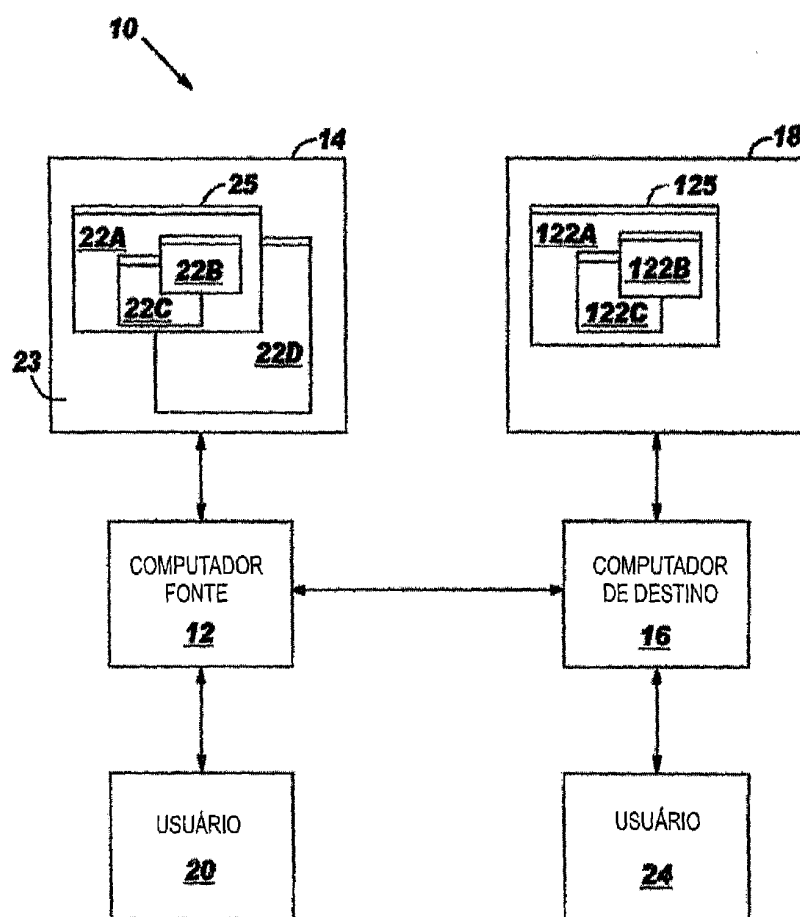


FIGURA 2

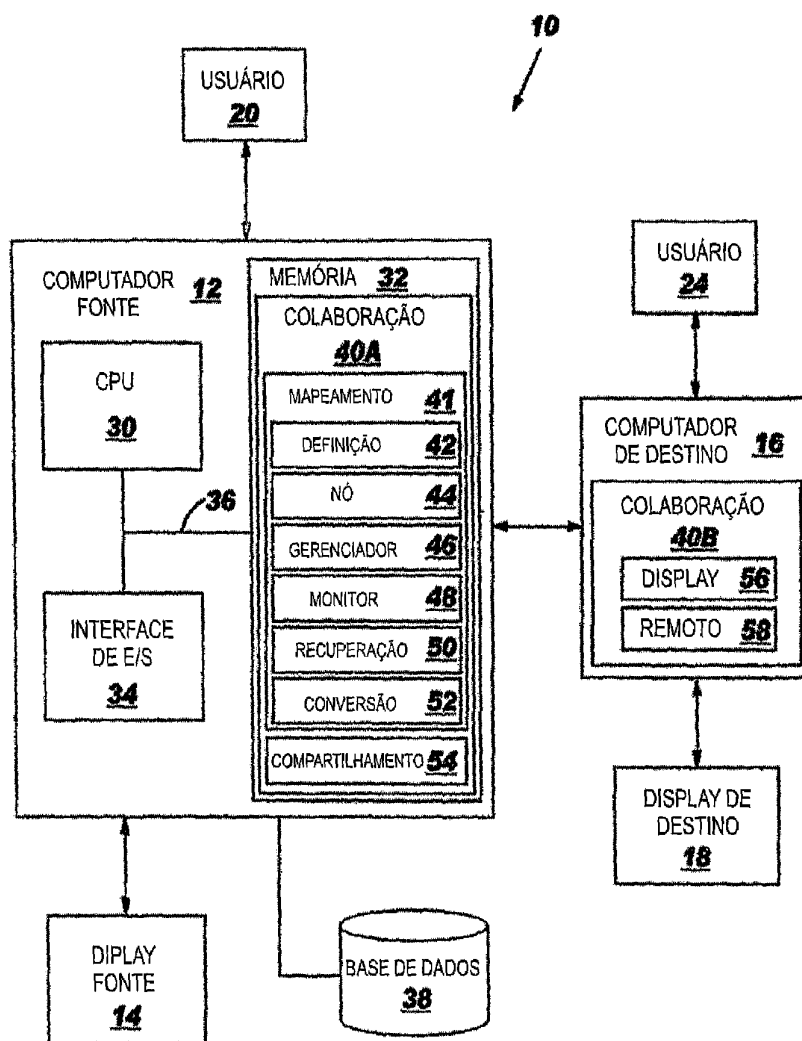


FIGURA 3

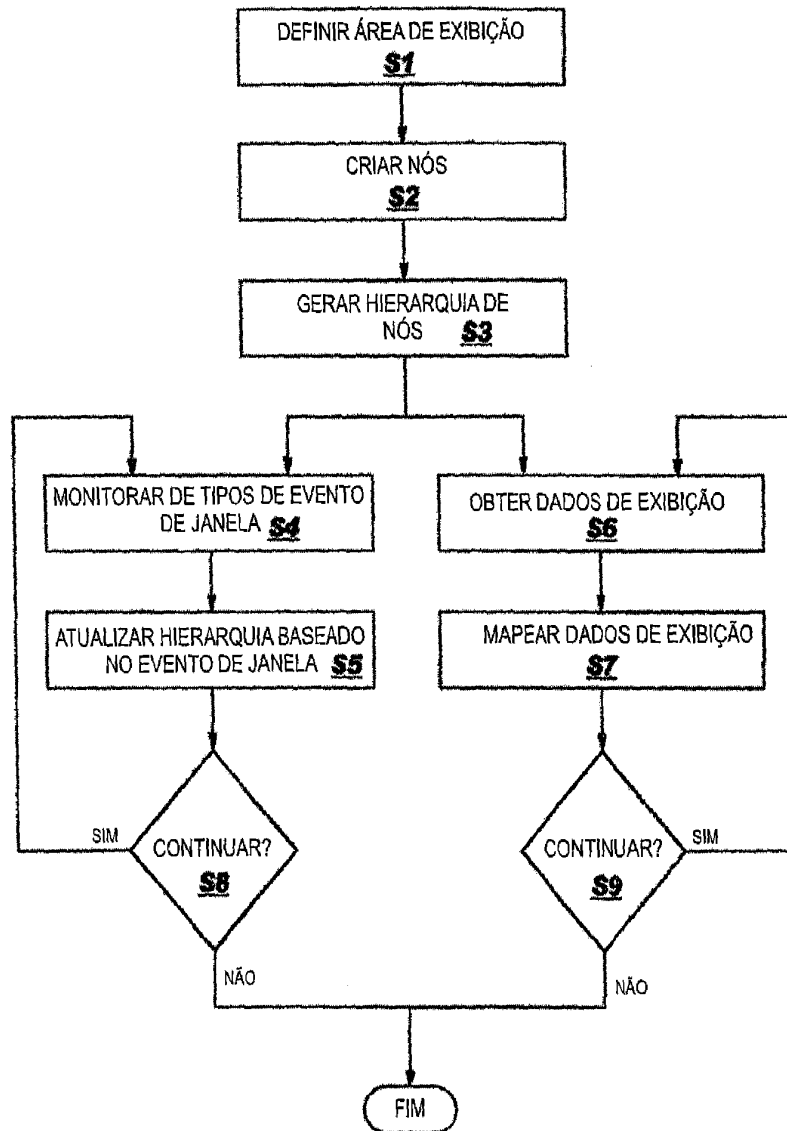


FIGURA 4

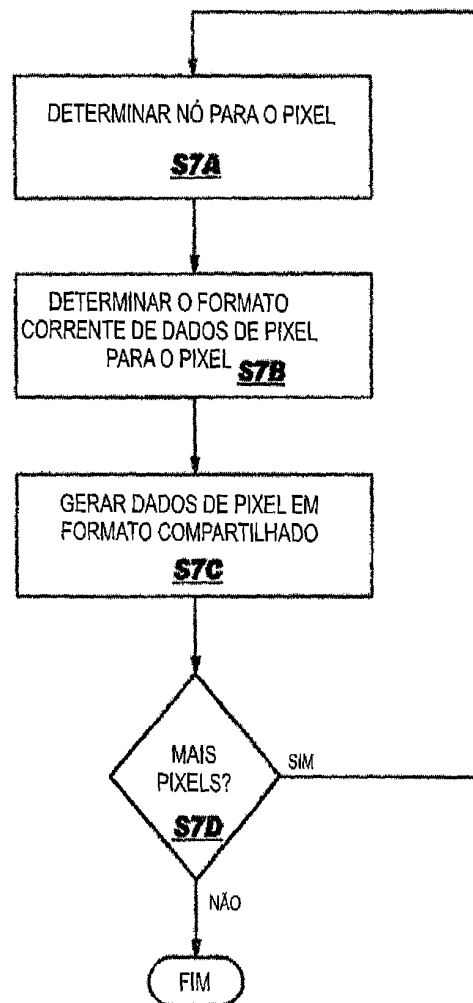


FIGURA 5

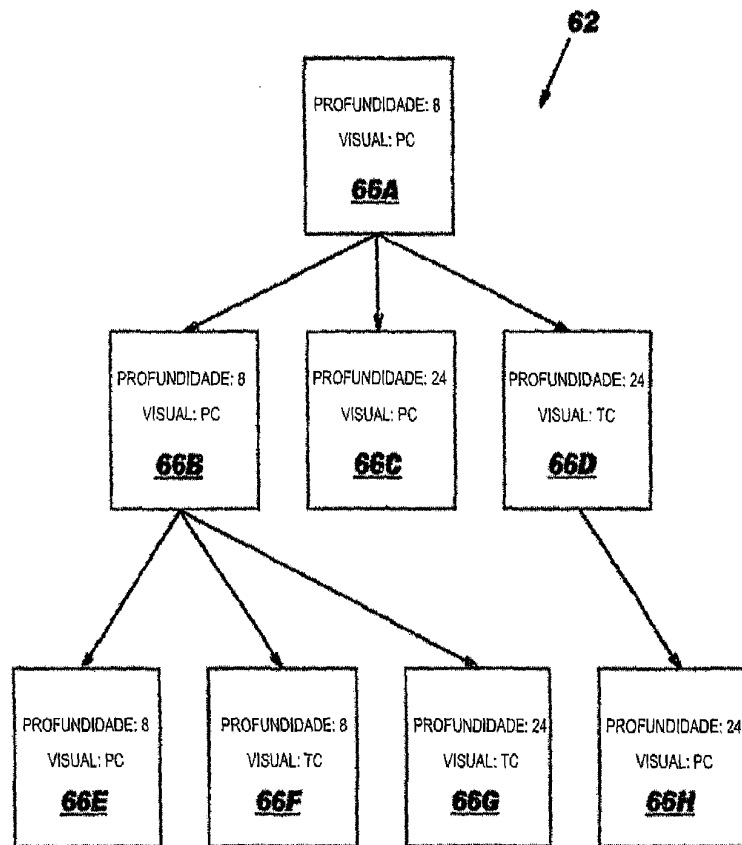


FIGURA 6

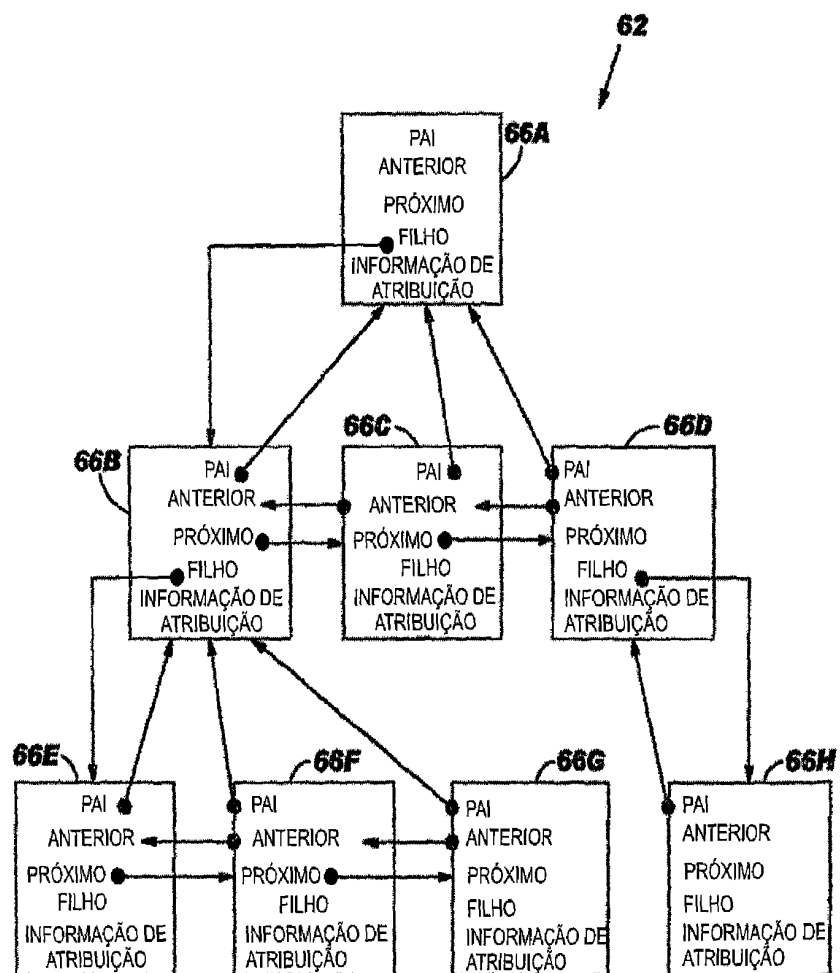


FIGURA 7

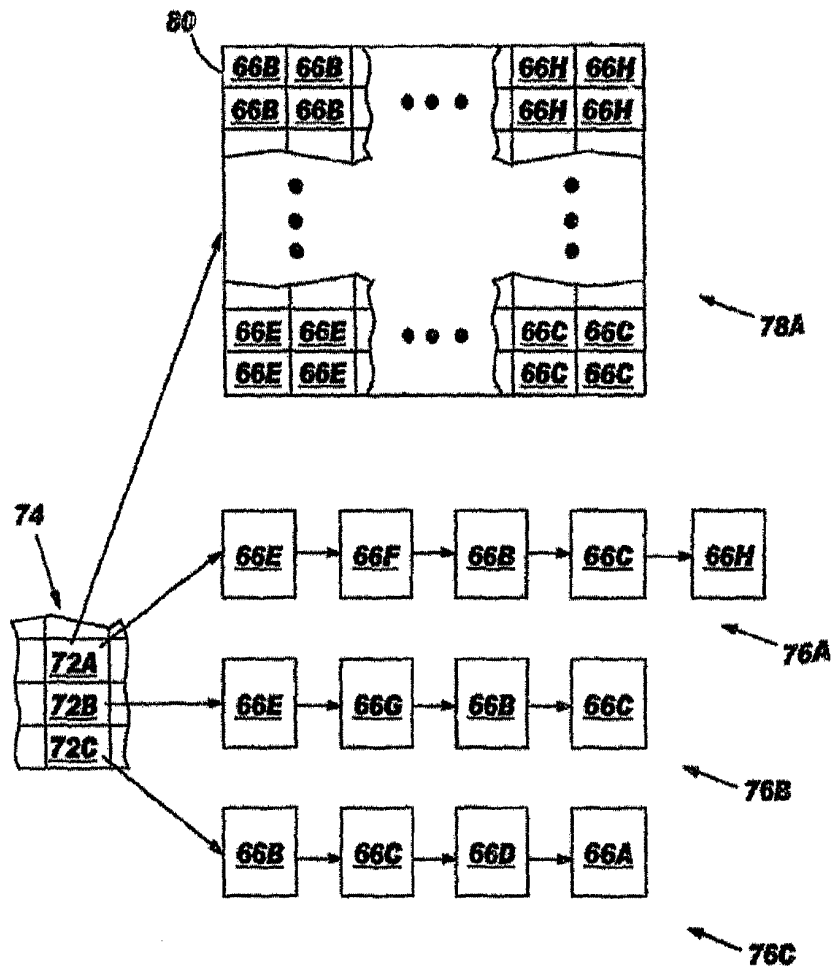


FIGURA 8