1
Introdução

Os dados bidimensionais são muito utilizados em várias aplicações de computação gráfica. Em geral, essas aplicações são desenvolvidas para permitir a visualização desses dados em tempo real, como por exemplo visualização de panoramas, imagens e terrenos. Um sistema de visualização em tempo-real deve assegurar uma taxa mínima e constante de quadros por segundo. A dificuldade de se realizar esta tarefa está intimamente relacionada com a quantidade de dados que se deseja visualizar. Primeiro, porque no processo de visualização é necessário que os dados sejam carregados em memórias de alta velocidade que possuem um pequeno poder de armazenamento. Segundo, porque é necessário efetuar cálculos geométricos e de iluminação num grande volume de dados para a visualização do objeto. O tempo destinado a realizar esses cálculos é bastante reduzido.

Para resolver o problema de armazenamento é necessário utilizar técnicas de gerenciamento de memória. Essas técnicas exploram as características dos objetos gráficos para determinar quais dados devem permanecer na memória e quais devem ser eliminados.

O problema relacionado ao pouco tempo de processamento é resolvido reduzindo a quantidade de dados que deve ser processada. Essa redução pode ser realizada a partir das técnicas de representação dos dados gráficos. Uma das técnicas mais utilizadas é a representação em multi-resolução.

O objetivo deste trabalho é desenvolver um sistema de gerenciamento de memória baseado no modelo de memória virtual para visualizar dados gráficos definidos em regiões retangulares. As imagens de satélite e panoramas virtuais são exemplos de dados gráficos com suportes retangulares. Esses objetos são representados em uma estrutura de multi-resolução. O sistema utiliza a forma de carregamento preditiva baseando-se no fato de que os objetos visualizados possuem coerência espacial e temporal.

1.1 Motivação

Hoje existem diversas aplicações da área de Computação Gráfica que necessitam trabalhar com uma grande quantidade de dados. Como a tecnologia de armazenamento não acompanha o crescimento dessa demanda, é necessário desenvolver técnicas de gerenciamento de memória que permitam que essa massa de dados seja processada e visualizada com a tecnologia atualmente disponível.

Os objetos gráficos 2D são muito utilizados em diversas áreas do conhecimento humano. Existem alguns tipos de objetos gráficos 2D que são naturalmente definidos a partir de uma grande quantidade de dados. A motivação de desenvolver um sistema de gerenciamento de memória para visualizar em tempo-real esses objetos vem dos seguintes fatores:

- São muito utilizados tanto na área acadêmica quanto na área comercial.
- Possuem grande quantidade de informação de textura associada a eles. Esta característica dificulta a operação de visualização em tempo-real.
- Podem ter complexidades geométricas muito diferentes. As imagens de satélite possuem uma geometria de complexidade simples, sua geometria é definida a partir de um plano. Os panoramas virtuais possuem uma geometria de complexidade intermediária, sua geometria é formada por cilindros e esferas. Os dados de terrenos formam uma geometria mais complexa.
- São visualizados em aplicações que possuem um sistema de gerenciamento de memória com uma hierarquia de armazenamento fixa e não escalável. Além disso esse sistema possui um bom sistema de predição. A conseqüência disto é que necessitam de máquinas com grande poder de armazenamento e processamento.

1.2 Problemas e Soluções

Para simular ou estudar um determinado fenômeno do mundo real através de processos computacionais é necessário alimentar o sistema com dados provenientes desse fenômeno. Em geral, quanto maior for a quantidade das amostras adquiridas, maior será a precisão dos resultados retornados pelo sistema.

Os sistemas computacionais são formados por uma hierarquia de unidades de armazenamento de dados e uma ou mais unidades de processamento. Essa hierarquia é organizada da unidade mais veloz para a unidade mais lenta. Cada nível dessa hierarquia é chamado de *estágio de armazenamento*. Por convenção, o estágio mais alto contém o dispositivo mais veloz e com menor capacidade de armazenamento. Os dados só podem ser processados pela unidade de processamento quando estão armazenados no estágio mais alto. As Figuras 1.1(a) e 1.1(b) ilustram duas hirarquias de armazenamento onde o estágio mais alto é representado pela memória RAM e pela memória de textura, respectivamente.

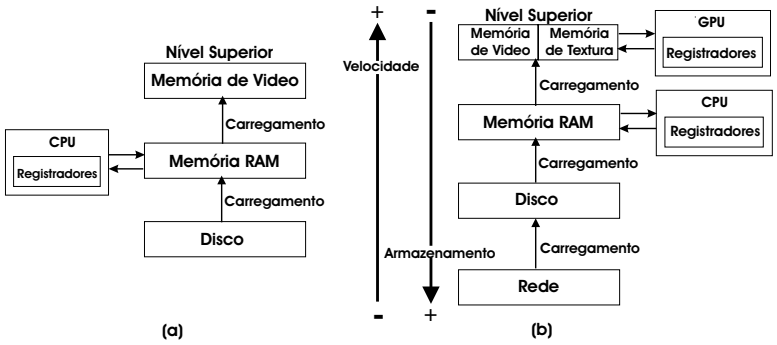

Figura 1.1: Hierarquia simples(a) e complexa(b) de processamento e armazenamento.

Por exemplo, a Figura 1.1(a) mostra como era a hierarquia dos primeiros sistemas de armazenamento. Estes sistemas possuíam uma arquitetura simples, formada basicamente por uma unidade primária (memória RAM) e uma ou várias unidades de disco e uma unidade de processamento (CPU). A memória principal e os registradores são os únicos dispositivos que podem ser acessados diretamente pela CPU. A parte gráfica era formada por uma controladora de vídeo, que não é ilustrada na figura, cuja única função era varrer uma determinada área da memória RAM e transferir os dados para a memória de vídeo. A Figura 1.1(b) mostra uma hierarquia mais complexa existente nos sistemas gráficos atuais. Observe que o sistema possui mais uma unidade de processamento chamada de unidade de processamento gráfico (GPU). As unidades de armazenamento que a GPU pode acessar são as memórias de vídeo e de textura. Note que a GPU é muito mais complexa que uma controladora de vídeo. Uma GPU é capaz de realizar vários tipos de cálculos geométricos, cálculos de iluminação, cálculos de mapeamento de textura e etc.

Devido ao problema de custo e limitações físicas, quanto mais alto estiver o estágio de armazenamento (nível superior), menor é a sua capacidade de armazenamento. O problema ocorre quando a quantidade de dados que se deseja processar não pode ser completamente armazenada neste estágio.

Para resolver este problema foram criados os sistemas de gerenciamento de memória. Estes sistemas baseiam-se no princípio de que a quantidade de dados necessária simultaneamente para se efetuar um determinado processamento pode ser armazenada no estágio mais alto. Um sistema de gerenciamento de memória é determinado a partir das seguintes características:

- A forma como a unidade de armazenamento é particionada e como os dados estão dispostos dentro dessas partições.
- Os algoritmos utilizados para determinar os dados que devem ser carregados para os estágios mais altos de armazenamento e os dados que devem ser retirados.
- A forma como os estágios estão interligados.

A primeira característica influencia a velocidade com que os dados podem ser transmitidos de um estágio para outro. A segunda, influencia a quantidade de operações de transferência de dados que deve ser feita durante algum tipo de processamento. A última característica determina a flexibilidade e a escalabilidade do sistema. As duas primeiras características têm um grande impacto no desempenho do sistema, e isto se reflete no desempenho do processamento. Assim, para que um sistema de gerenciamento de memória tenha um bom desempenho, é necessário maximizar a taxa de transferência de dados a partir de um bom planejamento da disposição dos dados nos níveis de armazenamento e desenvolver algoritmos que minimizem as operações de transferência de dados entre estes níveis.

Inicialmente os sistemas de gerenciamento de memória foram desenvolvidos com o objetivo de permitir que os sistemas operacionais executassem aplicações sem a necessidade de carregar na memória RAM (estágio mais alto) todos os dados e o código binário da aplicação. Os sistemas de gerenciamento de memória desenvolvidos para os sistemas operacionais eram caracterizados por terem poucos níveis de armazenamento, em geral dois níveis (RAM e Disco), e não serem escaláveis. O trabalho de Tanenbaum [4] aborda em detalhes a evolução e as tipos mais comuns de gerenciamento de memória utilizados em sistemas operacionais.

Em tais sistemas são implementados algoritmos de carregamento e liberação de dados genéricos. Normalmente o carregamento só é feito quando o dado é solicitado e a liberação é baseada no tempo de acesso. Essas operações são implementadas em hardware. Mesmo assim a maioria das aplicações perde desempenho durante a fase de processamento. A perda de desempenho é decorrente da generalidade com que os dados são tratados.

Para contornar os problemas de escalabilidade, flexibilidade e perda de desempenho, as aplicações passaram a desenvolver seus próprios sistemas de gerenciamento de memória. Baseado na natureza dos dados e na forma como eles são processados, pode-se dimensionar a quantidade de níveis de armazenamento, planejar a forma como os dados devem ser armazenados e desenvolver algoritmos de carregamento e liberação adequados para a forma como os dados são acessados pela aplicação. Como a forma de acesso aos dados é bem definida, é possível criar algoritmos que podem carregar os dados para os níveis mais altos de armazenamentos antes de serem solicitados pela aplicação. Esse tipo de algoritmo é chamado de *algoritmo preditivo*. Assim o processamento quase nunca é interrompido, o que aumenta o desempenho da aplicação.

Os primeiros sistemas de gerenciamento de memória foram desenvolvidos para permitir operações de cálculos numéricos como, por exemplo, resolver sistemas lineares e transformadas de Fourier, problemas de geometria computacional e busca sobre uma grande massa de dados. Nessas aplicações o tipo de processamento é bem comportado e isto facilita o desenvolvimento de algoritmos preditivos muito precisos. O trabalho de Jeffrey Vitter[38] discute vários tipos de algoritmos de carregamento e liberação e técnicas de armazenamento adequados para cada tipo de problema.

As aplicações gráficas são desenvolvidas não apenas para processar dados, mas também para permitir que eles sejam visualizados. A *latência* é o tempo que uma aplicação leva para responder a um evento realizado pelo usuário. As aplicações que respondem aos eventos sem que o usuário perceba nenhuma latência do sistema são chamadas de *aplicações em tempo-real*. Nas aplicações de visualização em tempo-real a latência está intimamente relacionada com os fatores perceptuais do olho humano. Esses fatores são baseados na freqüência de amostragem e na percepção de movimento do olho humano. A freqüência de amostragem do olho humano é de aproximadamente $55Hz$, ou seja, ele é capaz de capturar 50 amostras do ambiente por segundo. As freqüências acima $55Hz$ são chamadas de *freqüências de fusão* do olho humano. Devido a este fato é que os monitores

e as TVs têm uma taxa de varredura de $60Hz$ ou mais. Desta forma o olho humano não consegue perceber a formação da imagem na tela. Em relação à percepção de movimento é necessário assegurar uma taxa mínima e constante de trinta imagens por segundo para que o cérebro perceba um movimento contínuo dos objetos contidos numa seqüência de imagens. Assim, nas aplicações de visualização em tempo-real exige-se que sua latência máxima seja de $\frac{1}{30}s$ para poder simular objetos em movimento. Os exemplos mais conhecidos de visualização em tempo-real são os simuladores de vôo e de corridas de automobilismo.

O desenvolvimento de bons algoritmos de predição para as aplicações de visualização é uma tarefa bastante difícil devido a duas razões:

- Os dados que a aplicação deverá acessar no futuro dependem das ações realizadas pelo usuário e essas ações variam bastante com o tempo. A natureza não determinística dessas aplicações dificulta a tarefa de predizer o que deve ser carregado para os níveis mais altos de armazenamento. Aqui as decisões erradas de carregamento podem afetar o desempenho da aplicação e a qualidade da visualização.
- O pouco tempo disponível para o processamento impede a utilização de cálculos muito sofisticados para se encontrar a solução ótima de carregamento para cada ação realizada pelo usuário. Assim, é necessário criar heurísticas que aproximem a solução ótima.

O problema de predição será discutido com mais detalhes ao longo deste trabalho (Capítulo 7). Também será apresentado um sistema de gerenciamento de memória que possui uma estrutura hierárquica de armazenamento flexível e escalável.

Um outro problema é que para visualizar um determinado conjunto de dados é necessário realizar diversos tipos de processamento, como cálculos de projeção e iluminação, mapeamentos de textura e *culling*. Como o tempo para processar esses dados é muito limitado, é necessário restringir o campo de atuação desse processamento sobre o menor conjunto de dados possível.

Esse problema é resolvido a partir de técnicas de multi-resolução. Neste tipo de representação um conjunto de dados é representado e armazenado em vários níveis de detalhes, como mostra a Figura 1.2. Assim, durante o processo de visualização, um determinado nível de resolução é mostrado para o usuário. Do ponto de vista intuitivo, a escolha deste nível é baseada em características perceptuais dos seres humanos. Isto é, quando o usuário está observando o objeto de perto, é mostrado o nível de maior resolução. Por outro lado, quando o usuário vai se afastando

do objeto, são mostrados níveis de menor resolução. Do ponto de vista prático, esse nível é escolhido baseado na resolução do dispositivo de visualização (resolução do monitor). O nível de resolução escolhido é aquele onde cada um dos seus pontos é projetado em um pixel diferente da tela. Ou seja, busca-se um nível de resolução cuja razão entre número de pontos projetados e o número de pixels da tela seja mais próxima de um. Estes dois pontos de vista são equivalentes porque tanto o olho humano quanto o dispositivo de visualização possuem um número finito de sensores e pixels, respectivamente.

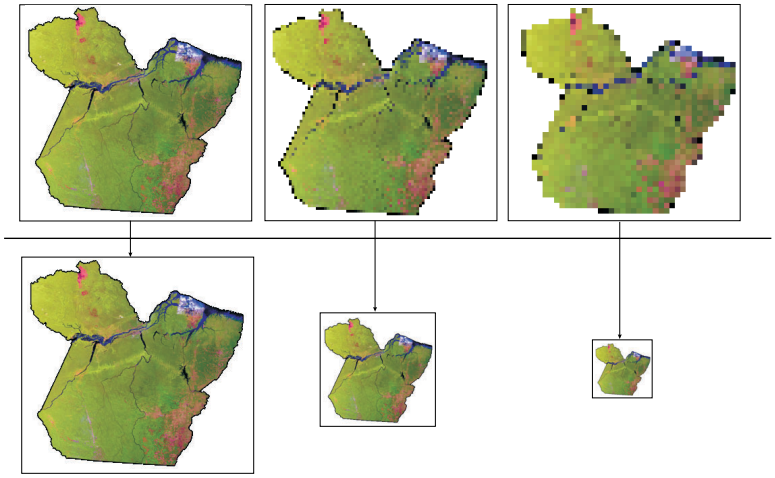

Figura 1.2: A parte superior ilustra uma imagem de satélite representada em diferentes níveis de detalhes. A parte inferior mostra a visualização desses níveis de detalhes na resolução correta.

A principal vantagem da representação em multi-resolução é que durante o processo de visualização uma quantidade praticamente constante de dados é processada. Isto ocorre porque quando o usuário quer ver detalhes de um objeto ele deve se aproximar do mesmo, diminuindo assim, o seu campo de visão. Desta forma, apenas uma pequena região pode ser visualizada por vez. À medida que o usuário se afasta do objeto uma maior região do mesmo é visualizada, no entanto, esta região pode ser mostrada com menos detalhes.

A desvantagem dessa técnica de representação é que exige uma grande quantidade de espaço de armazenamento. Isto ocorre devido ao armazenamento de informações redundantes nos vários níveis de resolução. Esta representação pode-se tornar inviável para grande quantidade de dados. Esse problema também será abordado neste trabalho.

1.3 Objetivos

O objetivo deste trabalho é desenvolver um sistema distribuído de gerenciamento de memória para visualizar em tempo-real objetos gráficos 2D, representados em multi-resolução.

O sistema de memória proposto neste trabalho possui as seguintes características:

- É baseado no modelo de memória virtual que cria um modelo de abstração, separando completamente o nível de memória lógica do nível de memória física. Esse modelo de abstração permite ao sistema gerenciar diversos tipos de unidades de armazenamento, como memória de vídeo, RAM, Disco, disco remoto e etc.
- Implementa cada estágio de armazenamento como módulos independentes que podem ser colocados ou retirados dependendo das necessidades da aplicação. Essas operações podem ser realizadas em tempo de execução ou em tempo de compilação.
- Divide os espaços de alocação das unidades de armazenamento em blocos de mesmo tamanho. Esta característica torna o sistema de paginação mais rápido.
- Permite acoplar um módulo que implementa o sistema de predição. No caso de visualização de dados, o sistema explora a coerência espacial e temporal para carregar dados que muito provavelmente serão necessários num futuro próximo.
- Independe do tipo da aplicação. Isso permite que o sistema possa ser utilizado em vários problemas relacionados com visualização.

Os dados bidimensionais são abordados de acordo com o conceito de objeto gráfico [21]. Essa abstração permite lidar com os vários tipos de dados de uma forma única. Esse modelo teórico é essencial para tornar o sistema de gerenciamento de memória independente do tipo de dado e conseqüentemente, do tipo da aplicação.

Um objeto gráfico é definido por um suporte geométrico que contém informações topológicas e geométricas, e uma função de atributos que definem as propriedades deste objeto. Neste nível de abstração, os dados bidimensionais são chamados de *objetos gráficos 2D*. Os objeto gráficos 2D são caracterizados por terem o domínio do seu suporte geométrico definido no plano. Os exemplos mais conhecidos são as superfícies e as imagens.

Os objetos gráficos 2D utilizados neste trabalho possuem suporte geométrico contínuo, cujo domínio seja uma área retangular, e a função de atributos é definida por um mapeamento de textura. Os objetos gráficos 2D escolhidos para testar o sistema de gerenciamento de memória foram as panoramas virtuais e as imagens de satélite. As aplicações de visualização de terrenos não foram exploradas devido à grande complexidade dos algoritmos de síntese de imagem. Essa complexidade se reflete na vasta literatura que existe sobre o assunto. A figura 1.3 mostra um exemplo de cada um desses objetos gráficos.

Figura 1.3: Objetos gráficos 2D: (a)Imagem de satélite; (b) Panorama virtual.

A metodologia adotada neste trabalho está dividida em duas fases: representação e visualização. A fase de representação consiste das operações de decomposição e representação em multi-resolução adaptativa do objeto gráfico 2D. A operação de decomposição tem o objetivo de dividir os dados em unidades menores. Esta operação possibilitará ao sistema de gerenciamento de memória carregar apenas os dados que são necessários. A estrutura em multi-resolução adaptativa permite que o objeto gráfico seja representado em vários níveis de detalhes de forma concisa. Isto é feito a partir do controle do nível de redundância de dados entre os níveis de detalhes.

A fase de visualização compreende os sistemas de gerenciamento de memória e síntese de imagem. O sistema de gerenciamento de memória tem o objetivo de carregar os dados que provavelmente serão utilizados pelo sistema de síntese e eliminar os dados que não são mais necessários. O sistema de gerenciamento de memória é único para todos os objetos gráficos. O sistema de síntese realiza as operações de visualização. Este sistema é específico porque cada tipo de objeto gráfico pode utilizar técnicas diferentes de visualização. A Figura 1.4 dá uma visão geral do fluxo de dados e da ordem como as operações são realizadas.

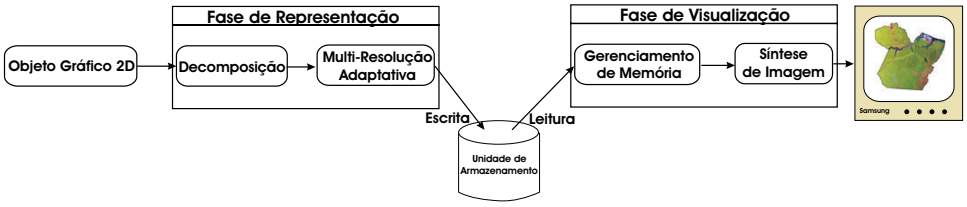

Figura 1.4: A seqüência de operações realizadas para visualizar um objeto gráfico 2D.

1.4 Contribuições

Este trabalho mostra que, utilizando uma boa técnica de gerenciamento de memória, é possível processar uma grande massa de dados. Também é defendido que esse processamento pode ser realizado utilizando uma tecnologia de fácil acesso no mercado. Aqui, o problema de gerenciamento de memória é discutido dando ênfase ao problema de visualização em tempo-real.

As principais contribuições deste trabalho estão na forma como o problema de gerenciamento de memória é tratado. O problema é abordado a partir do conceito de objeto gráfico. Assim, foi possível criar um sistema que é independente do tipo de dado utilizado na aplicação.

A arquitetura do sistema foi inspirada no conceito de memória virtual. Este conceito permite que os problemas relacionados com gerenciamento de memória sejam completamente separados da aplicação. Além disso, é proposta uma arquitetura flexível e escalável. A flexibilidade permite ao sistema incorporar vários estágios de armazenamento. Os estágios foram projetados para suportar vários dispositivos físicos de mesma natureza. Essa propriedade permite que o sistema seja distribuído.

Durante a fase de execução as aplicações fazem acesso a um ou vários dispositivos de armazenamento. Os dados acessados devem ser carregados nesses dispositivos para que a aplicação possa realizar o seu processamento. Este trabalho estuda o problema de carregamento antecipado de dados. O desenvolvimento de boas soluções para esse problema é essencial para obter um bom desempenho da aplicação e do sistema de gerenciamento como um todo. Desta forma, é proposta uma nova técnica de predição para aplicações de visualização em tempo-real. A tarefa de carregamento é realizada levando em consideração o comportamento dos acessos da aplicação e a velocidade com que os dados podem ser carregados para o dispositivo desejado. Esses fatores podem variar bastante com o tempo. A técnica apresentada neste

trabalho consiste em monitorar estas variações e tomar as decisões mais adequadas para cada instante do tempo, ou seja, é proposto um algoritmo adaptativo para carregamento antecipado de dados.

1.5 Estrutura

Esta tese foi estruturada em nove capítulos e um apêndice. O segundo capítulo mostra o estado da arte do que se tem feito sobre visualização em tempo-real de objetos gráficos 2D. Apresenta e analisa os trabalhos mais importantes na área, mostrando suas vantagens e suas desvantagens. Também é mostrado como este trabalho difere dos demais.

O terceiro capítulo aborda o conceito de objeto gráfico. Mostra também exemplos de objetos gráficos utilizados neste trabalho.

O quarto capítulo mostra todos os detalhes de construção da estrutura em multi-resolução adaptativa. Para construir essa estrutura é utilizada a técnica de decomposição e de multi-resolução. A representação em multi-resolução é realizada utilizando o conceito de erro acumulado.

O quinto capítulo mostra como fazer a visualização dos objetos gráficos representados a partir da representação em multi-resolução adaptativa. Esse capítulo mostra as técnicas de visualização para imagens de satélite e panoramas virtuais.

No sexto capítulo são apresentados a arquitetura e o funcionamento do sistema de gerenciamento de memória proposto neste trabalho. Será mostrado como cada estágio de armazenamento está estruturado, como é feita a transferência de dados e a comunicação entre cada estágio, como estão estruturados os sistemas de carregamento, liberação e ausência de páginas e como um endereço virtual é transformado em um endereço real.

O sétimo capítulo apresenta e discute o problema de predição, bem como a solução proposta neste trabalho. Também são mostradas as regras para carregamento, liberação e ausência de páginas que devem ser utilizadas com o objetivo de garantir coerência entre essas operações.

No oitavo capítulo são discutidos os resultados obtidos com os sistemas de predição e gerenciamento de memória em aplicações de visualização de panoramas virtuais e imagens de satélite. Durante os testes vários fatores são levados em consideração como hardware, movimento de câmera e tamanho dos blocos de memória.

O nono capítulo inicia dando uma visão geral dos problemas relacionados com sistema gerenciamento de memória para aplicações em

tempo-real. Em seguida, é feita uma análise crítica das soluções propostas neste trabalho. Com base nessas críticas são propostas novas direções de pesquisas para trabalhos futuros.

O apêndice A apresenta o modelo de memória virtual. Mostra como é a arquitetura e o funcionamento dos sistemas de gerenciamento de memória baseados neste modelo. Esse apêndice é útil para quem não está familiarizado com os conceitos de memória virtual.