

1

Introdução

Sistemas de visualização vêm sendo utilizados em diversas áreas da indústria e do campo científico. Dentre essas áreas, CAD (Computer Aided Design), visualização científica e realidade virtual fazem uso desses sistemas para a inspeção detalhada de modelos de engenharia, análise de dados científicos e implementação de sistemas com alto grau de imersão.

Com o crescente desenvolvimento dos processadores gráficos (GPUs), a visualização em tempo real já é uma realidade, o que viabiliza uma gama de novas aplicações. As demandas atuais de tais aplicações incluem altas taxas de renderização, baixo tempo de resposta aos comandos do usuário, alta resolução e qualidade das imagens geradas em tempo real, realismo na representação do mundo virtual e simulações físicas e de inteligência artificial com alta precisão e complexidade.

Para garantir tal qualidade de visualização e simulação são utilizados modelos 3D normalmente compostos por um número muito grande de primitivas geométricas. O realismo das imagens geradas é alcançado por meio do uso de modelos de iluminação bastante complexos que incluem efeitos naturais como reflexão, refração e sombra, entre outros. A demanda por altas resoluções tem sido atendida através da construção de sistemas com múltiplos projetores. Para preencher a tais requisitos computacionais é necessário um poder de processamento gráfico sem precedentes.

São duas as maneiras de melhorar o desempenho de sistemas de visualização:

- implementação de técnicas de aceleração, que visam lançar mão da menor quantidade de processamento gráfico possível para a geração de uma imagem correta ou parcialmente correta;
- aumento do poder de processamento bruto do sistema, o que é feito empregando-se ou estações de trabalho altamente especializadas ou agrupamentos de PCs.

As estações de trabalho acima mencionadas utilizam diversas linhas de produção gráfica que operam em paralelo. Essas linhas de produção são

altamente integradas e têm acesso a uma memória compartilhada por meio de barramentos que possuem grande largura de banda. O maior problema dessa solução é o seu alto custo — da ordem de milhões de dólares — o que a torna inviável para diversos projetos na área de visualização.

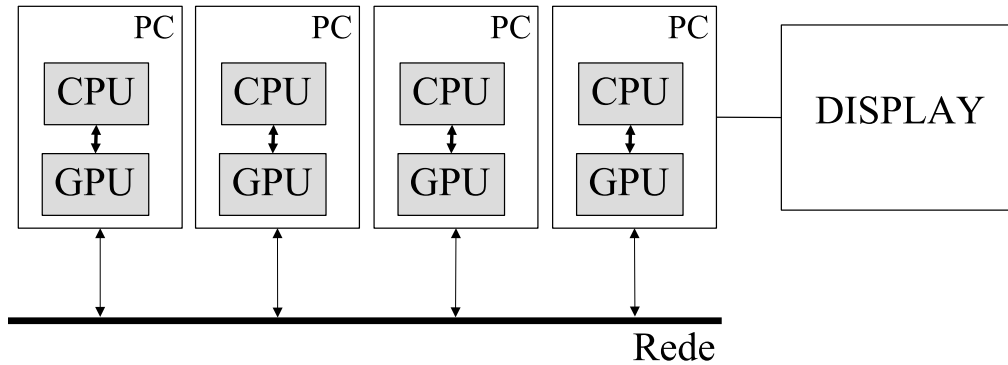


Figura 1.1: Diagrama de um agrupamento de PCs para renderização.

Agrupamentos de PCs (*PC clusters*) são conjuntos de PCs de baixo custo interligados por uma rede de alta velocidade. Esses agrupamentos são utilizados para resolver problemas bastante complexos, o que é feito com o uso de algoritmos paralelos e distribuídos.

O uso de agrupamentos de PCs para a visualização em tempo real tem sido uma alternativa bastante estudada recentemente. Neles, cada PC possui um processador gráfico (GPU). Entre esses computadores, que também são chamados *nós* do agrupamento de PCs, são divididas diferentes tarefas, entre elas o controle da renderização distribuída, a renderização, a composição de renderizações parciais e a visualização do resultado final da renderização. Um diagrama de um agrupamento de PCs para visualização encontra-se na Figura 1.1.

O uso desses agrupamentos para renderização apresenta vantagens bastante atraentes em relação ao emprego de estações de trabalho especializadas, entre elas:

- relação custo-benefício muito mais vantajosa do que a observada no uso de estações de trabalho especializadas;
- maior flexibilidade no acompanhamento da evolução da tecnologia dos aceleradores gráficos, que têm progredido num passo bem mais acelerado do que a Lei de Moore [1]; ¹

¹A Lei de Moore prediz que o poder computacional dos processadores comuns dobra a cada 18 meses.

- cada PC possui um barramento próprio entre sua CPU e sua GPU — ao contrário do barramento de estações especializadas, que é compartilhado pelas linhas de produção gráfica — o que possibilita que a largura de banda agregada entre a CPU e a GPU cresça linearmente com o número de PCs.

Um dos grandes desafios no uso de um agrupamento de PCs é a construção de um sistema escalável, ou seja, um sistema cujo poder de processamento cresça proporcionalmente ao número de processadores. Em um ambiente distribuído como o de um agrupamento de PCs, é necessário fazer uma distribuição explícita dos dados pertinentes à visualização entre os processadores gráficos [17, 25], problema que não existe em um sistema com memória compartilhada. O volume de dados que deve ser transferido pode facilmente sobrecarregar a rede que conecta os PCs, causando perda de desempenho. Em vista disso, algumas opções no projeto da arquitetura são cruciais para a construção de um sistema escalável.

Devido à grande quantidade de dados que devem trafegar pela rede, é necessário utilizar algoritmos de renderização distribuída que operem sob a largura de banda disponível na rede. É desejável também que os dados sejam comprimidos sempre que possível, no intuito de reduzir a carga na rede. As demandas por altas taxas de renderização e baixa latência de resposta ao usuário são especialmente difíceis de se atender quando a largura de banda disponível é restrita.

Dividir igualmente a computação entre os processadores é outro grande desafio para os algoritmos paralelos e distribuídos. Devem ser desenvolvidos algoritmos de balanceamento de carga para manter todos os processadores com um volume igual de trabalho, evitando assim que haja ociosidade, a qual se traduz em desperdício de recursos computacionais que poderiam ser utilizados em outras tarefas.

Com relação às arquiteturas de renderização distribuída, Molnar et al. [7] as classificaram com base em qual estágio da linha de produção gráfica ocorre a ordenação das primitivas para o tratamento da visibilidade. Três classes de arquiteturas foram concebidas com base neste critério: arquiteturas com ordenação no início (*Sort-First*), no meio (*Sort-Middle*) e no fim (*Sort-Last*). Dentre elas, as arquiteturas com ordenação no início e com ordenação no fim são implementáveis em agrupamentos de PCs.

Na arquitetura com ordenação no início, a tela é dividida em um conjunto de ladrilhos disjuntos (normalmente com forma retangular), os quais são distribuídos aos nós do agrupamento. Cada processador fica responsável por desenhar todas as primitivas que intersectam o volume

de visão determinado pelo seu ladrilho. No fim da renderização de cada ladrilho, basta mostrá-lo no dispositivo de visualização, pois os ladrilhos não se intersectam. Essa é a arquitetura que utiliza a menor largura de banda da rede, porém ela é bastante suscetível ao desbalanceamento de carga proveniente da distribuição desigual de primitivas sobre a tela. Para resolver tal problema, devem ser desenvolvidos algoritmos para particionar a tela em ladrilhos com igual carga, de forma a manter todos os processadores do agrupamento ocupados.

Já na arquitetura com ordenação no fim, cada processador fica responsável por desenhar uma parte do modelo completamente. Depois, a parte da tela em que cada processador desenhou primitivas deve ser lida e enviada a um processador de composição, que resolve a visibilidade de cada fragmento com base nas sub-imagens de todos os processadores. Para que a resolução de visibilidade seja possível em outro processador, devem ser enviadas as informações de cor e de visibilidade (profundidade ou componente *alpha*, dependendo do algoritmo). Essa arquitetura apresenta ótima escalabilidade em termos do número de primitivas desenhadas, pois cada primitiva é desenhada por apenas um processador. No entanto, ela requer maior largura de banda da rede, pois cada processador deve enviar dados de cor e visibilidade (possivelmente de uma tela cheia) a um processador de composição, o que pode facilmente sobrecarregar a rede que conecta os processadores do agrupamento.

Este trabalho apresenta um novo sistema de renderização distribuída destinado a agrupamentos de PCs que controlam um único monitor para a visualização. Esse sistema disponibiliza uma infra-estrutura para a implementação de algoritmos de renderização distribuída, a qual servirá de base para testar três arquiteturas: uma arquitetura com ordenação no início, uma arquitetura com ordenação no fim para renderização volumétrica e uma arquitetura híbrida que tenta combinar as vantagens da ordenação no início e da ordenação no fim.

Neste sistema é feita uma extensão da linha de produção gráfica convencional através de uma linha de produção gráfica distribuída. Cada computador do agrupamento utiliza múltiplas linhas de execução para paralelizar as operações feitas na CPU, na GPU e na rede que interliga os computadores do agrupamento.

Um algoritmo de balanceamento de carga é proposto para arquiteturas que dividem a tela em ladrilhos disjuntos, o qual equilibra a carga baseado nos tempos de renderização dos ladrilhos. Esse algoritmo é geral, podendo ser empregado em aplicações com gargalo tanto no estágio da geometria

quanto no estágio da rasterização, além de se comportar bem com cenas dinâmicas. É também proposta uma técnica de distribuição de trabalho entre os nós de renderização do agrupamento, visando minimizar o tempo de ociosidade das GPUs causado pelo desbalanceamento de carga.

Um novo algoritmo de partição do modelo entre os computadores do agrupamento é proposto para a arquitetura híbrida. A partição é feita implicitamente nos nós do agrupamento, evitando que um computador seja sozinho o responsável por particionar o modelo inteiro a cada quadro.

No Capítulo 2 é feita uma recapitulação dos trabalhos relacionados às áreas de renderização em tempo real e de renderização distribuída. O Capítulo 3 descreve a linha de produção gráfica distribuída. Os algoritmos de balanceamento de carga e distribuição de trabalho propostos para a arquitetura com ordenação no início e o algoritmo de partição para a arquitetura híbrida com ordenação no início e no fim são descritos no Capítulo 4. O Capítulo 5 apresenta os resultados experimentais dos sistemas e algoritmos propostos. Finalizando, no Capítulo 6, são apresentadas as conclusões e idéias para trabalhos futuros.