

Aplicação de Paralelismo na Síntese de Imagens Através do Método da Radiosidade

Gladimir V. Guimarães Baranoski
Leonardo de Carvalho Garcia

Informatics Institute - CPGCC - UFRGS - Caixa Postal 15064 - Porto Alegre - Brasil
FAX: (051) 336-5576 - E-mail: gladimir@inf.ufrgs.br garcia@inf.ufrgs.br

Abstract

This paper presents the first application of the Cooperative Processes Software (CPS) on radiosity methods using a parallel strategy based on software rather than hardware. The application is described and the preliminary results as well as the projections regarding future efforts in this area are pointed out.

1. Introdução

O emprego de imagens geradas por computador vem recebendo um destaque significativo nos últimos anos. Estas imagens têm importantes aplicações envolvendo a simulação de fenômenos naturais e a produção de sequências de animação com finalidades educacionais ou de entretenimento. Os esforços nesta área da computação gráfica visam aumentar o grau de realismo das imagens e desenvolver processos de síntese mais acurados do ponto de vista da aplicação de leis físicas.

O método da radiosidade [1] destaca-se entre os métodos de síntese de imagens por apresentar um tratamento mais acurado da simulação das reflexões de luz. Entretanto, este método apresenta um tempo de processamento significativo. Com a finalidade de reduzir este tempo vários pesquisadores estão desenvolvendo estudos propondo o emprego de arquiteturas paralelas em aplicações envolvendo este método. Estes estudos propõem alternativas paralelas baseadas em "hardware" [2-3].

Este trabalho propõe a aplicação de paralelismo na síntese de imagens através do método da radiosidade, utilizando uma ferramenta de "software" conhecida como "Cooperative Processes Software" (CPS) [4-6]. Será descrita a aplicação proposta de paralelismo nas diferentes etapas do processo de síntese e serão apresentados e analisados os resultados já obtidos.

2. Fundamentos do Método da Radiosidade

O método da radiosidade foi introduzido por Goral *et al.* [1] e Nishita e Nakamae [7]. É um método orientado para a síntese de imagens por computador, baseando-se em princípios de conservação de energia e transferência de energia calorífica por radiação. Estes princípios são utilizados há bastante tempo em engenharia e foram adaptados de forma a modelar a interação de energia radiante luminosa (luz) entre superfícies refletoras.

Os principais aspectos que diferenciam o modelo de iluminação apresentado pelo método da radiosidade dos modelos anteriores ao seu surgimento são os seguintes:

- modelar a reflexão de luz entre objetos e/ou superfícies refletoras difusas (as quais compõem a maioria dos ambientes reais);
- incluir efeitos de mistura de cores ("color blending") causados pelas reflexões difusas;
- empregar fontes de luz de área finita e
- ser independente da posição do observador.

Este método consiste basicamente no cálculo das *radiosidades* dos N setores, ou “patches”, que compõem as superfícies de um ambiente fechado. A radiosidade de um “patch” é a soma da luz emitida e da luz refletida por ele (1). Para calcular as radiosidades é necessário especificar tanto as relações geométricas entre os “patches” que compõem o ambiente fechado, os *fatores de forma*, quanto os aspectos de emissão luminosa, *reflectividade* e *emissividade*, associados a cada “patch”.

$$B_j = E_j + \rho_j \sum_{i=1}^N B_i F_{ij} \quad (1)$$

onde:

B_j = radiosidade da superfície j , representa a taxa total de energia radiante luminosa que deixa a superfície j em termos de energia por unidade de tempo e por unidade de área;

E_j = emissividade da superfície j , representa a taxa de energia radiante luminosa que deixa a superfície j por unidade de tempo e por unidade de área;

ρ_j = reflectividade da superfície j , representa a fração de energia radiante luminosa incidente, que é refletida pela superfície j de volta para o ambiente;

B_i = radiosidade da superfície i ;

F_{ij} = fator de forma, representa a fração de energia radiante luminosa que deixa a superfície i e atinge a superfície j .

Os fatores de forma representam o ponto crítico do método da radiosidade, já que além de influenciarem diretamente na exatidão das radiosidades, o seu cálculo corresponde a uma parcela bastante significativa do tempo total gasto na síntese das imagens. Existem diversos métodos de cálculo de fatores de forma. Neste trabalho optou-se pela utilização do *Método do Diferencial Paramétrico (MDP)* [8]. Este método trata convenientemente superfícies curvas e apresenta um elevado grau de exatidão no cálculo dos fatores de forma.

Uma vez calculados os fatores de forma e arbitrados os valores de emissividade e reflectividade de cada “patch” é montado o sistema de N equações lineares (2) que caracteriza o balanço de energia do ambiente fechado. A fim de se obter a radiosidade completa de cada “patch”, é necessário montar e resolver este sistema de equações para cada banda de interesse do sistema de cores que se deseja empregar. Neste trabalho utilizou-se o método conhecido como “*LU Decomposition*” [9] na resolução dos sistemas de equações.

$$\begin{bmatrix} 1 - \rho_1 F_{11} & -\rho_1 F_{12} & -\rho_1 F_{1N} \\ -\rho_2 F_{21} & 1 - \rho_2 F_{22} & -\rho_2 F_{2N} \\ \vdots & \vdots & \vdots \\ -\rho_N F_{N1} & -\rho_N F_{N2} & 1 - \rho_N F_{NN} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_N \end{bmatrix} \quad (2)$$

Na etapa seguinte, emprega-se um algoritmo de remoção de elementos ocultos para determinar os elementos do ambiente visíveis ao observador. Neste trabalho, optou-se pelo algoritmo de “*ray-casting*” [10]. Na etapa final, as radiosidades obtidas são normalizadas e interpoladas de forma a permitir a determinação da intensidade de cada “pixel” da imagem. Estas duas últimas etapas constituem o processo de “*rendering*” propriamente dito das imagens.

3. Descrição da Ferramenta

O *CPS* é uma ferramenta que reúne um conjunto de facilidades para a programação, depuração e execução de aplicações paralelas e foi originalmente concebida para a utilização no multiprocessador *ACP II* [4-5]. Criado

no Fermi National Accelerator Laboratory, o projeto *ACP III/CPS* destina-se basicamente ao processamento dos dados obtidos em experiências de física experimental de altas energias [4], as quais apresentam uma demanda computacional elevada. O *CPS* destina-se à implementação de paralelismo de alta granularidade. Os objetos que executam em paralelo são definidos por processos de usuário com capacidade de comunicação e sincronização entre si. Estes processos podem ser executados paralelamente quando cada processo é alocado individualmente a uma máquina ou concorrentemente quando um conjunto de processos é alocado a uma mesma máquina.

O *CPS* foi desenvolvido para plataformas compatíveis com *UNIX* atualmente incluindo “drivers” para os sistemas *SUN*, *Apollo* e *Silicon Graphics* (estações de trabalho) e *VAX* (“mainframe” executando *VMS* ou *ULTRIX*) [5]. Conseqüentemente, aplicações paralelas podem ser desenvolvidas e executadas em redes locais heterogêneas. O processo de desenvolvimento e depuração de aplicações é bastante flexível utilizando-se os compiladores, depuradores e outras ferramentas próprias de cada máquina.

O *CPS* é composto por dois módulos, um para a programação e outro para a execução, depuração e monitoração de aplicações. O primeiro módulo é constituído por um conjunto de primitivas básicas escritas em linguagem *C* e oferece os mecanismos necessários para a implementação de processos cooperativos, tais como sincronização (barreiras), chamada remota a procedimento, transferência de mensagens, inicialização e término de processos. O segundo módulo, denominado “*Job Manager*”, é o processo responsável, entre outras tarefas, pela gerência da aplicação incluindo inicialização e término de processos, alocação de recursos junto ao sistema operacional, redirecionamento das operações de E/S de processos em execução em máquinas remotas.

O projeto de uma aplicação *CPS* implica necessariamente três etapas: particionamento do algoritmo seqüencial em uma solução paralela composta de tarefas funcionalmente diferentes; codificação dos programas que farão parte da aplicação; e confecção de um arquivo chamado “*Job Description File*” que descreve uma configuração de execução da aplicação indicando número de processos a criar para cada tarefa e um mapeamento de alocação de processos para processadores. A parte mais complexa deste processo de projeto da aplicação paralela é o particionamento da solução em várias tarefas de execução paralela. Além da definição de cada tarefa, devem ser concebidas interfaces que determinem a interação entre tarefas de uma forma cooperativa.

4. Arquitetura da Aplicação

O emprego de paralelismo será efetuado nas três etapas mais importantes do processo de síntese das imagens:

- Etapa 1: no cálculo dos fatores de forma;
- Etapa 2: na resolução dos sistemas de equações lineares e
- Etapa 3: no “rendering” da imagem.

Estas etapas possuem características que proporcionam um elevado grau de paralelismo. O cálculo dos fatores de forma de cada “patch” do ambiente é independente do cálculo dos fatores de forma dos demais “patches”. A resolução do sistema de equações para cada banda de interesse do sistema de cores também é efetuada de forma independente. Do mesmo modo, o algoritmo empregado na etapa de “rendering” calcula a intensidade de cada “pixel” de forma separada. A estratégia para a aplicação de paralelismo nesta última etapa consistirá basicamente na divisão da tela do dispositivo de exibição em regiões, sendo os cálculos correspondentes a cada região realizados em paralelo.

A aplicação proposta foi particionada nas seguintes tarefas:

- Tarefa 1: Processo “cliente” que inicializa estruturas de dados globais e coordena a execução de etapas de cálculo através de chamadas a subrotinas remotas implementadas nos processos definidos pelas tarefas 2,3 e 4.

- Tarefa 2: Implementa o serviço de cálculo de fatores de forma. Dependendo do número de estações de trabalho disponíveis podem ser criadas várias réplicas deste serviço.
- Tarefa 3: Implementa o método “LU Decomposition” de resolução de sistemas lineares de equações. Em um primeiro momento, são criados apenas três processos implementando a tarefa 3. Cada um fica responsável pela resolução de um sistema de equações correspondente a uma banda de interesse do sistema de cores. Esta abordagem, no entanto, limita as possibilidades de um ganho maior de desempenho, já que serão utilizados apenas três processadores da rede, permanecendo os demais subutilizados durante esta etapa do processo. A alternativa que será pesquisada e empregada em trabalhos futuros será a exploração de paralelismo interno ao método utilizado. Desta forma, um número maior de processadores será empregado e obter-se-á um melhor desempenho.
- Tarefa 4: Calcula a intensidade de cada “pixel” de uma determinada região do dispositivo de exibição de acordo com o mapeamento das coordenadas espaciais correspondentes à cena para as coordenadas de dispositivo. A tarefa 4 deve ser executada por vários processos.
- Tarefa 5: Realiza a exibição da imagem. Esta tarefa é executada por apenas um processo que recebe as intensidades dos “pixels” a serem exibidos diretamente dos processos servidores da tarefa 4.
- Tarefa 6: Dedicada à gravação do arquivo de fatores de forma e do arquivo imagem.

5. Resultados Obtidos

Na fase atual da implementação já foram obtidos resultados relativos ao cálculo dos fatores de forma. Os testes, cujos resultados são apresentados neste trabalho, foram realizados em uma base de “hardware” formada por uma rede composta por cinco estações de trabalho SUN Sparc 2. Os resultados obtidos correspondem à utilização exclusiva da rede. Em outras palavras, os processadores estavam alocados à execução dos serviços descritos neste trabalho e não estavam sendo utilizados simultaneamente por outros usuários.

O gráfico mostrado na Figura 1 apresenta os ganhos de desempenho em relação à aceleração, “*speed-up*”, correspondentes à execução sequencial e às execuções paralelas do algoritmo de cálculo dos fatores de forma. Estes valores são confrontados com o “*speed-up*” ideal. Os ganhos de desempenho obtidos são praticamente lineares em relação ao número de processadores utilizados. Tomando-se como exemplo o emprego de três processadores, o “*speed-up*” obtido foi de 2,84.

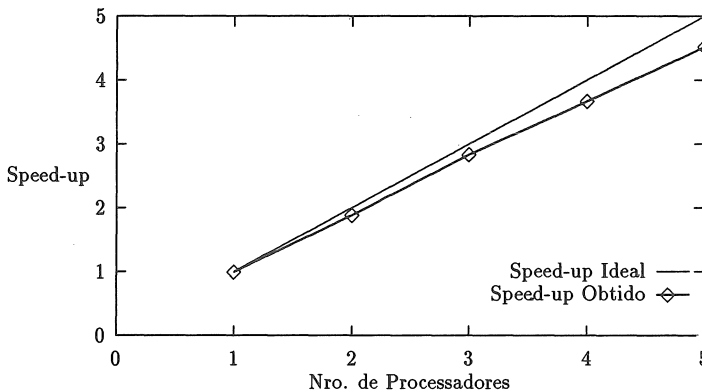


Figura 1. Speed-up x número de processadores.

Além da análise referente ao “speed-up”, foi efetuada a comparação dos tempos correspondentes às execuções seqüencial e paralela das etapas do processo de síntese. Esta comparação baseou-se no tempo dispendido na síntese de uma imagem correspondente a um ambiente dividido em 992 “patches”. Esta imagem, apresentada na figura 2, foi exibida em uma estação de trabalho SUN 4/330 “true-color” com uma resolução de 1152x900 “pixels”.

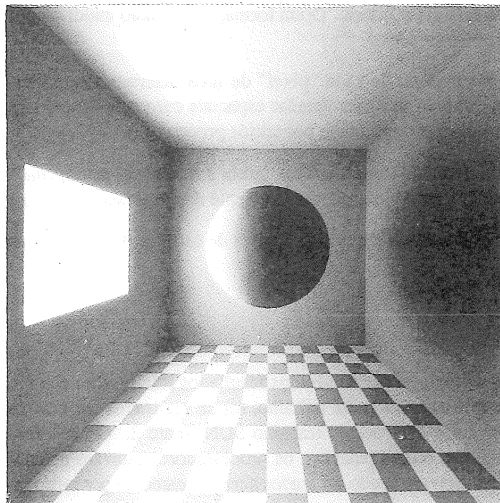


Figura 2. Imagem gerada empregando-se o método da radiossidade.

Os tempos de execução seqüencial de cada uma das três etapas descritas anteriormente são apresentados na Tabela 1. Estes valores ressaltam a importância do cálculo dos fatores de forma no processo de síntese das imagens, sendo responsável por 81,29% do tempo total. Por este motivo, neste trabalho, foi priorizada a paralelização desta etapa.

Tabela 1. Tempos de execução seqüencial das etapas do processo de síntese.

Etapa	Tempo de Execução	Fração do tempo total (%)
1	176m 36s	81,29
2	35m 56s	16,54
3	4m 43s	2,17

Os tempos de execução paralela das etapas do processo de síntese são apresentados na Tabela 2. Embora os valores correspondentes às etapas 2 e 3 sejam valores estimados, baseados nos valores de “speed-up” referentes à etapa 1, é indiscutível o ganho de desempenho que pode ser obtido aplicando-se a estratégia de paralelismo proposta. Na versão final da implementação da aplicação proposta visando a síntese de imagens em um ambiente de produção, as etapas 2 e 3 também serão implementadas por tarefas paralelas. Embora estas etapas não apresentem frações de tempo tão significativas quanto a etapa 1, a aplicação de paralelismo nestas aumentará a eficiência global do processo. Ainda, como imagens diferentes de uma mesma cena não necessitam de um recálculo dos

fatores de forma, a paralelização das etapas 2 e 3 torna-se, além de conveniente, necessária para que se possa obter imagens diferentes mais rapidamente.

Tabela 2. Tempos de execução paralela das etapas do processo de síntese.

Etapa	Tempo de execução	Nro. de Processadores	Speed-up
1	39m 1s	5	4,52
2	12m 1s	3	2,84*
3	1m 3s	5	4,52*

* valores estimados

6. Conclusões

A estratégia proposta, por ser baseada em “software”, apresenta certas vantagens relevantes. Primeiramente não implica a utilização de uma base de “hardware” específica, permitindo o emprego de uma gama variada de equipamentos sem problemas de portabilidade. Além disso, permite que a aplicação proposta seja executada em máquinas de altíssimo desempenho, como o *ACP II*.

Na aplicação apresentada foi utilizada uma abordagem “full-matrix” do método da radiosidade. Esta abordagem tem como característica a execução seqüencial das etapas do processo de síntese. Por este motivo a estratégia proposta restringiu-se ao emprego de paralelismo dentro de cada etapa. A utilização de uma abordagem do tipo refinamento progressivo [12] poderá permitir também a exploração de paralelismo temporal. Neste caso obter-se-á um maior grau de desempenho pela conjugação harmoniosa de dois tipos de paralelismo:

- paralelismo real obtido em cada estágio através da replicação dos serviços e
- paralelismo temporal obtido com a sobreposição dos tempos de execução dos serviços executados em estágios diferentes.

Conforme mencionado anteriormente, os resultados apresentados neste trabalho correspondem à utilização de uma rede de nodos dedicados ao processamento da aplicação proposta. Resultados bem mais modestos são obtidos quando os recursos da rede são compartilhados por outros usuários. Além disto, a performance de aplicações interativas executadas em um nodo utilizado por um processo de aplicação paralela apresenta uma significativa degradação no desempenho, uma vez que os dois tipos de processos consomem muitos ciclos do processador. Como o *CPS* foi originalmente projetado para executar em um ambiente dedicado (multiprocessador *ACP II*), esse problema não é solucionado na atual versão da ferramenta. Portanto, a execução da aplicação proposta em uma rede local de processadores não dedicados exigirá a utilização de um mecanismo de balanceamento dinâmico e adaptativo de carga, como o proposto por Garcia [11]. Este mecanismo considera aspectos tais como capacidade de processamento de cada processador, tempo de execução dos serviços e gerenciamento de carga imposta por outros usuários da rede.

Os ganhos de desempenho já obtidos e esperados justificam o emprego da estratégia proposta. Entretanto, para a utilização de redes com grande número de nodos, torna-se necessário o projeto cuidadoso de tarefas e serviços a fim de que se possa obter o melhor desempenho possível. Em trabalhos futuros serão apresentadas análises relativas ao comportamento da aplicação em redes heterogêneas de alto potencial de processamento formadas por um grande número de processadores.

Referências

1. C.M. Goral, K.E. Torrance, D.P. Greenberg and B. Battaile. "Modeling the Interaction of Light Between Diffuse Surfaces". *Computer Graphics*, Computer Graphics, 18(2) pp. 213-222 (1984).
2. M.Feda and W.Purgathofer. "Progressive Refinement Radiosity on a Transputer Network", In: *Proceedings of the Second EUROGRAPHICS Workshop on Rendering*, (1991).
3. A.G.Chalmers and D.J.Paddon. "Parallel Processing of Progressive Refinement Radiosity Methods", In: *Proceedings of the Second EUROGRAPHICS Workshop on Rendering*, (1991).
4. T.Nash, H.Areti, R.Atac, J.Biel, A.Cook, J.Deppe, M.Edel, M.Fischler, I.Gaines, R.Hance, D.Husby, M.Isely, E.Miranda, T.Pharm, T.Zmuda, E.Eitchen, G.Hockney, H.B.Thacker, P.Mackenzie and D. Toussaint. "Software for Event Oriented Processing on Multiprocessr Systems", In: *Proceedings of the Symposium on Recent Developments, Processors and Software for High-Energy Physics*. Guajanato, Mexico (1984).
5. T.Nash, H.Areti, R.Atac, J.Biel, A.Cook, J.Deppe, M.Edel, M.Fischler, I.Gaines, R.Hance, D.Husby, M.Isely, E.Miranda, T.Pharm, T.Zmuda, E.Eitchen, G.Hockney, H.B.Thacker, P.Mackenzie and D. Toussaint. "High Performance Parallel Computers for Science: New Developments at the Fermilab Advanced Computer Program", In: *Proceedings of the Workshop on Computational Atomic and Nuclear Physics at One Gigaflop*, (1988).
6. Fermi National Accelerator Laboratory. "ACP Cooperative Processes Software - User's Manual", Universities Research Association, (1989).
7. T.Nishita and E.Nakamae. "Continuous Tone Representation of Three Dimensional Objects Taking into Account of Shadows and Interreflections". *Computer Graphics*, Computer Graphics, 19(3) pp. 23-30 (1985).
8. G.V.G.Baranoski. "The Parametric Differential Method: An Alternative to the Calculation of Form Factors", In: *Proceedings of EUROGRAPHICS Conference and Exhibition*, (1992).
9. W.H. Press, B.P.Flannery, S.A.Teukolsky and W.T.Vetterling. *Numerical Recipes in C The Art of Scientific Computing*, Cambridge University Press, 1988.
10. J.D.Foley, A. Van Dan, S.K. Feiner and J.F. Hughes. *Fundamentals of Interactive Computer Graphics*, Reading, Addison-Wesley, 1990.
11. L.C.Garcia. "Balanceamento de Carga em Sistemas Paralelos Fracamente Acoplados", CPGCC - UFRGS, Porto Alegre, 1992. (Dissertação de mestrado em andamento)
12. M.Cohen, S.E.Shenchang, J.R.Wallace and D.P.Greenberg. "A Progressive Refinement Approach to Fast Radiosity Image Generation". *Computer Graphics*, Computer Graphics, 22(4) pp. 75-84 (1988).

Agradecimentos

Os autores deste trabalho gostariam de agradecer aos pesquisadores do *Laboratório de Física Experimental de Altas Energias - Centro Brasileiro de Pesquisas Físicas (LAFEX/CBPF)* e do *Fermi National Accelerator Laboratory (FermiLab)* por terem viabilizado a utilização do CPS e aos colegas Valdir Belmonte, Roberto Amboni e Marinho Pilla Barcelos do CPGCC-UFRGS pela sua colaboração na revisão deste trabalho.