

Paralelização da interpolação *Inverse Distance Weighted (IDW)* utilizando CUDA

Andrey Rodrigues*
Tecgraf

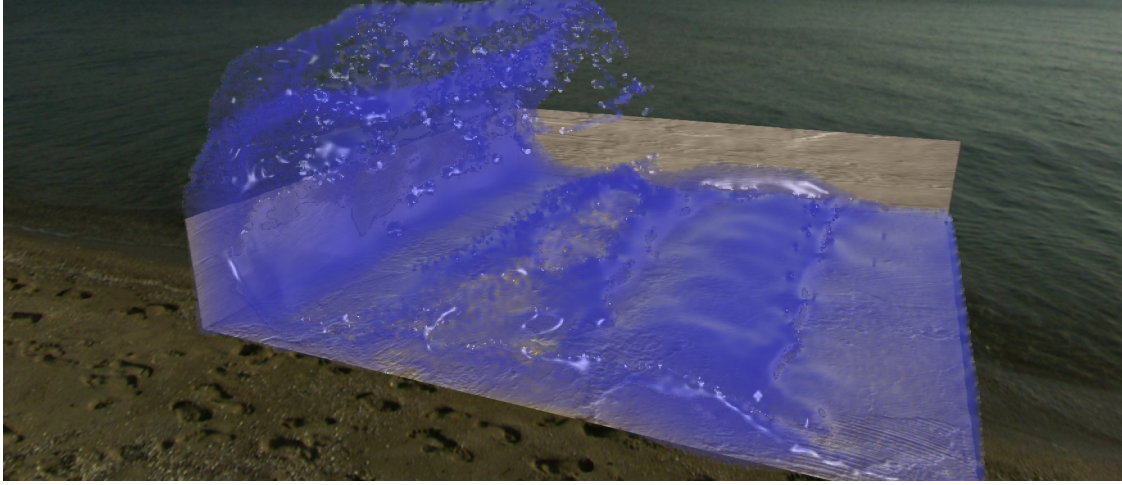


Figura 1: Imagem final gerada utilizando 128K partículas de uma simulação SPH

Resumo

Este documento descreve a implementação do algoritmo IDW utilizando CUDA. Foram feitas 3 implementações do mesmo algoritmo: Uma em CPU; uma estratégia simples em CUDA utilizando somente memória global; e a última também em CUDA utilizando memória compartilhada e uma estratégia de *tiling*. Foram feitos experimentos para comparar o tempo de execução entre os algoritmos e foi atingido um speedup de até xxx para os casos de teste.

Keywords: CUDA, IDW, algorithm, shared memory

1 Introdução

A interpolação espacial é um método fundamental em geociências, onde um número de dados conhecidos, como alturas de terreno por exemplo, são utilizados para prever dados desconhecidos no mesmo modelo. O custo computacional destes algoritmos cresce de acordo com os dados de entrada conhecidos da interpolação e o número de dados que necessitam ser computados. Geralmente implementações seriais destes algoritmos se tornam muito custosas em relação ao tempo quando a quantidade de dados é grande.

O IDW, é o algoritmo mais comumente utilizado em interpolações espaciais, dado sua implementação bem simples e direta. Shepard, propôs uma função para a interpolação IDW bi-dimensional e é comumente utilizada. No método de Shepard, todos os valores conhecidos são utilizados para computar um valor interpolado, e cada

cálculo por sua vez é independente de todos os outros, e, portanto, é facilmente paralelizável.

O advento da programação em placas gráficas nas últimas décadas, promoveu uma chamada natural para que pesquisadores desenvolvessem soluções para problemas conhecidos em paralelo. O CUDA (Compute Unified Device Architecture), é uma plataforma para programação em paralelo e um modelo de programação criado pela NVIDIA, que permite explorar o poder de multiprocessamento de GPU's para solucionar problemas computacionais complexos.

Neste trabalho foram desenvolvidas 3 versões do algoritmo IDW: uma sequencial em CPU, para fins de controle do tempo de resposta em algoritmo; Uma versão simples em GPU, para identificar os ganhos com o poder de paralelismo das placas gráficas; E outra versão em GPU explorando o cache manual dos dados de entrada em memória shared visando diminuir o acesso à memória global da GPU que é conhecida por ser lenta.

2 Metodologia

Neste trabalho foi utilizado o método de Shepard simples para o cálculo da interpolação. O cálculo pode ser definido como: Seja z_i o valor conhecido no ponto D_i , para calcular o ponto interpolado P utilizamos a função de interpolação

$$f(P) = \begin{cases} \frac{\sum_{i=1}^N w_i z_i}{\sum_{i=1}^N w_i}, & \text{se } d_i \neq 0 \\ z_i, & \text{se algum } d_i = 0 \end{cases} \quad (1)$$

onde $w_i = d(P, D_i)^{-u}$ sendo d uma métrica de distância, u uma expoente definido pelo usuário.

*e-mail:arodrigues@tecgraf.puc-rio.com

3 Resultados e discussões

4 Conclusão e trabalhos futuros

Referências

- BAGAR, F., SCHERZER, D., AND WIMMER, M. 2010. A layered particle-based fluid model for real-time rendering of water. In *Proceedings of the 21st Eurographics Conference on Rendering*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, EGSR'10, 1383–1389.
- KROG, O. E., AND ELSTER, A. C. 2012. Fast gpu-based fluid simulations using sph. In *Proceedings of the 10th International Conference on Applied Parallel and Scientific Computing - Volume 2*, Springer-Verlag, Berlin, Heidelberg, PARA'10, 98–109.
- LORENSEN, W. E., AND CLINE, H. E. 1987. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.* 21, 4 (Aug.), 163–169.
- MÜLLER, M., SCHIRM, S., AND DUTHALER, S. 2007. Screen space meshes. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '07, 9–15.
- ROSENBERG, I. D., AND BIRDWELL, K. 2008. Real-time particle isosurface extraction. In *Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games*, ACM, New York, NY, USA, I3D '08, 35–43.
- VAN DER LAAN, W. J., GREEN, S., AND SAINZ, M. 2009. Screen space fluid rendering with curvature flow. In *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games*, ACM, New York, NY, USA, I3D '09, 91–98.
- WILLIAMS, B. W. 2008. Smooth surface reconstruction from particles. In *In Proceedings of SIGGRAPH '08: ACM SIGGRAPH 2008 Papers. Association of Computing Machinery, ACM, Press.*