

1 Introdução

Simulação de partículas tem sido usada em uma grande gama de aplicações. Em animação, sistemas de partículas são usados para aumentar o realismo, especialmente em simulações de fenômenos naturais como água, fogo e fumaça [13,18,4]. Em análises numéricas, simulações de partículas têm sido cada vez mais usadas para analisar modelos sujeitos a grandes deformações, como fluídos e materiais granulares [11,3]. Independente da aplicação, resultados com precisão só são obtidos com grande número de partículas.

Com a rápida evolução das placas gráficas, soluções baseadas na GPU têm tido desempenho superior a implementações feitas na CPU. A maior eficiência no acesso à memória, combinada com o aumento da facilidade de programação, tem transformado a GPU em uma ótima opção para resolver computacionalmente problemas genéricos. Por isso, muitos pesquisadores já exploraram a GPU para a implementação de sistemas de partículas [17,10,9].

Este trabalho tem como objetivo apresentar um sistema de partículas executado inteiramente na GPU. Este sistema deve detectar e responder a colisão entre as partículas e a colisão das partículas contra o ambiente de simulação. O sistema também precisa ser numericamente estável, mesmo em ambientes confinados. Também é desejável que as partículas possuam diâmetros variáveis.

Para atender estes objetivos, é apresentado um novo simulador de partículas inteiramente implementado na GPU. Como a simulação é executada inteiramente no processador gráfico, a transferência de dados entre a GPU e a CPU é completamente eliminada. Ao contrário de propostas anteriores, este sistema consegue fazer simulações de partículas numericamente estáveis, mesmo em ambientes confinados, incluindo detecção e resposta a diversas colisões simultâneas entre partículas. Em placas gráficas atuais, o sistema proposto é capaz de simular mais de um milhão de partículas a taxas iterativas.

Este trabalho também propõe uma estratégia para a modelagem dos obstáculos que definem a cena onde ocorre a simulação. Isso permite a criação de

diferentes cenários sem a necessidade da recodificação dos *shaders* (programas que são executados pela GPU).

O restante desta dissertação é organizado da seguinte maneira. O próximo capítulo descreve alguns trabalhos relacionados. O Capítulo 3 descreve uma implementação na CPU do sistema proposto. O Capítulo 4 discute programação genérica na GPU. O Capítulo 5 descreve o sistema e sua implementação na GPU. Os resultados obtidos são tratados no Capítulo 6 e finalmente as conclusões e sugestões para trabalhos futuros são apresentados no Capítulo 7.