



<b>Evento</b>	Salão UFRGS 2013: SIC - XXV SALÃO DE INICIAÇÃO CIENTÍFICA DA UFRGS
<b>Ano</b>	2013
<b>Local</b>	Porto Alegre - RS
<b>Título</b>	Combinando Kd-trees e Packed Memory Arrays para Acelerar Detecção de Colisão de Partículas
<b>Autor</b>	AFFONSO DICK NETO
<b>Orientador</b>	JOAO LUIZ DIHL COMBA

A simulação física de partículas é um dos temas de estudo da computação gráfica. O cálculo da interação entre partículas requer computar a colisão para cada par de partículas, resultando em um algoritmo de complexidade quadrática. Quando o número de partículas é grande, o cálculo de colisão torna-se muito custoso. Estruturas de dados espaciais baseadas em particionamento do espaço podem ser utilizadas para reduzir os pares a serem testados e tornar o processo mais eficiente. Um particionamento possível do espaço utiliza um hiperplano para cortar o espaço em dois semi-espacos: positivo e negativo. Este particionamento binário é aplicado recursivamente seguindo o paradigma de divisão e conquista, de forma a reduzir o número de partículas em cada semi-espaco para a metade do número de partículas. Este processo é armazenado no formato de uma árvore binária, e é concluído quando um número aceitável de partículas reside em cada semi-espaco, ou quando uma profundidade máxima da árvore foi atingida. Quando os hiperplanos são paralelos aos planos principais essa estrutura é chamada de árvore k-dimensional (*kd-tree*).

Um dos aspectos que afeta a performance da *kd-tree* é associado ao custo de acesso e gerenciamento de memória. Para reduzir a latência de acessos a memória, caches são utilizadas para guardar dados frequentemente usados. No problema de teste de colisão de partículas, é necessário ter as informações das partículas disponíveis, se possível na cache. Quando uma das partículas a ser testada colisão não está na cache (*cache miss*), a informação deve ser buscada na memória principal, causando um atraso no processamento. O gerenciamento de memória na *kd-tree* é ainda mais complicado em cenas dinâmicas, onde partículas se movimentam, o que faz com que a *kd-tree* precise ser reorganizada, causando uma grande movimentação de memória.

Neste trabalho propomos o uso de uma estrutura de dados utilizada na área de banco de dados, chamada de *Packed Memory Array (PMA)*, para armazenar os dados da *kd-tree*. O PMA mantém os dados em uma mesma região de memória para reduzir o *cache miss*, combinado com uma área livre que flexibiliza a inserção/remoção de novas partículas, e portanto diminuindo a movimentação de memória quando os dados se movimentam. Como o PMA é uma estrutura baseada em árvore binária similar a *kd-tree*, foi feita uma fusão das duas estruturas. O resultado é uma estrutura híbrida que faz a classificação espacial dos elementos e ao mesmo tempo faz o gerenciamento de memória, evitando os problemas citados acima.