

**Andrea Zotovici**  
andrea.zotovici@gmail.com

**Ricardo Nakamura**  
ricardo.jogos@gmail.com

Universidade São Judas Tadeu (USJT) e FATEC, Brazil.

**Abstract:** We present the results of a systematic research about the application of adaptive techniques in collision detection, especially in virtual reality systems. The research showed us that some algorithms have a better performance due to the adaptive technique applied. The aim is apply the adaptive technology in collision detection to improve performance and precision in simulation systems to the health area.

**Keywords:** Adaptive Techiques; Collision Detection; Virtual Reality.

**Resumo:** Apresentamos os resultados de uma pesquisa sistemática sobre a aplicação de técnicas adaptativas em detecção de colisão, especialmente em sistemas de realidade virtual. A pesquisa mostrou que alguns dos algoritmos têm melhor desempenho devido a aplicação de técnicas adaptativas. O objetivo é aplicar a tecnologia adaptativa em detecção de colisão para melhorar o desempenho e precisão em sistemas de simulação para a área de saúde.

**Palavras Chaves :** Tecnologia Adaptativa; Detecção de Colisão; Realidade Virtual.

## 1 INTRODUÇÃO

Atualmente, muitas aplicações utilizam a realidade virtual para simulação de elementos com os quais seria difícil ou impossível de interagir e observar com frequência no mundo real.

Conforme [1], no ensino de Medicina, a simulação de procedimentos como cirurgia endovascular, quando utiliza cadáveres humanos, apesar de proporcionar alto grau de realidade e simulação completa, porém não é validado e usa apenas uma vez o material endovascular, o que gera alto custo. Por outro lado, a realidade virtual simula a mudança das especificações do paciente, possui alto grau de realismo e não há limitação de re-uso do material [1].

No ensino de biologia, os alunos somente podem visualizar células e micro-organismos em laboratório, com auxílio de equipamentos especializados. Um ambiente de realidade virtual poderia ser utilizado para simular essas células e micro-organismos, ampliando-os para que possam ser visualizados sem auxílio dos equipamentos especializados do laboratório e com mais detalhes que as imagens bidimensionais usadas nos livros convencionais.

No entanto, os alunos não aprendem apenas visualizando, é importante interagir com o modelo e, ao explorá-lo, é importante o realismo na reação ao estímulo. Para essa interatividade, faz-se necessária a detecção de colisão, ou seja,

o contato entre o objeto de interesse e o usuário (seja através de técnicas de manipulação direta ou indireta).

Para a detecção de colisão de objetos sólidos e não deformáveis, já existem muitas técnicas que aumentam a eficiência da detecção e reduzem o tempo de processamento. Porém, essas técnicas não são eficientes em tempo real quando aplicadas em objetos deformáveis devido ao processo de reestruturação[2].

A revisão sistemática foi realizada com o objetivo de identificar as técnicas de detecção de colisão que utilizam tecnologia adaptativa e verificar.

As próximas seções deste artigo estarão organizadas da seguinte maneira: na seção 2 serão discutidos conceitos básicos sobre detecção de colisão, os quais são abordados em trabalhos que agregam adaptatividade. Na seção 3 serão discutidas as técnicas de detecção de colisão que utilizam adaptatividade. Na seção 4 são discutidas aplicações que utilizam técnicas adaptativas de detecção de colisão em sistemas da área de saúde. Por fim, na seção 5 estarão as considerações finais.

## 2 DETECÇÃO DE COLISÃO

Por motivos de otimização de desempenho quanto ao tempo de execução, o processo de detecção de colisão pode ser dividido em duas fases *broad* e *narrow*. Na primeira fase, denominada *broad*, são identificados os objetos próximos, candidatos a colisão, por meio do teste de intersecção de seus volumes envoltórios. Esses objetos são enviados para a próxima fase, denominada *narrow*, para que seja aplicado um algoritmo mais detalhado para detecção de colisão.

Entre os algoritmos de volumes envoltórios estão incluídos o uso de caixas alinhadas aos eixos, caixas orientadas, esferas e politopos, detalhados a seguir:

Caixas alinhadas aos eixos (Axis Aligned Bounding Box- AABB) - os objetos candidatos a colisão são envolvidos por uma caixa na qual a normal de cada face é paralela aos eixos do sistema de coordenadas, como ilustra a Figura 1;

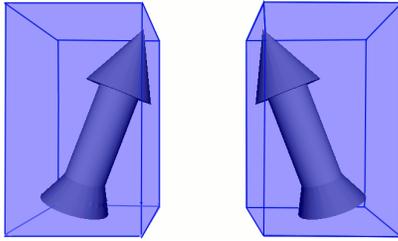


Figura 1 Caixas Alinhadas

Caixas orientadas (Oriented Bounding Box-OBB) – caixas com orientação de acordo com o objeto que envolve, conforme ilustra a Figura 2;

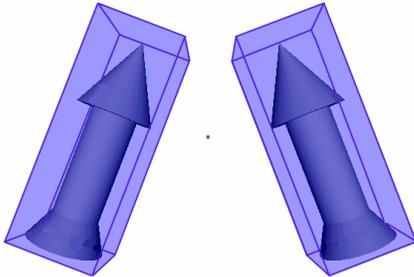


Figura 2 Caixas orientadas

Esferas– os objetos candidatos a colisão são envolvidos por uma esfera, conforme ilustra a Figura 3;

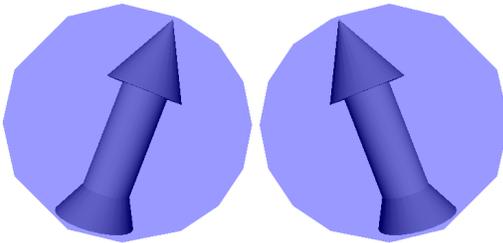


Figura 3 Esferas

Politopos com  $k$  orientações discretas (DOP-Discrete Oriented Polytope), são objetos convexos definidos por um pequeno conjunto fixo de  $k$  direções ( $D_1, \dots, D_k$ ) e uma tupla  $(d_1, \dots, d_k) \in R^k$  de escalares[3]. Como por exemplo, para 6DOP há seis direções:  $D1=(-1,0,0)$ ,  $D2=(0,-1,0)$ ,  $D3=(0,0,-1)$ ,  $D4=(1,0,0)$ ,  $D5=(0,1,0)$  e  $D6=(0,0,-1)$ [4].

Os politopos podem criar um volume envoltório mais justo ao objeto, como ilustrado pela Figura 4.

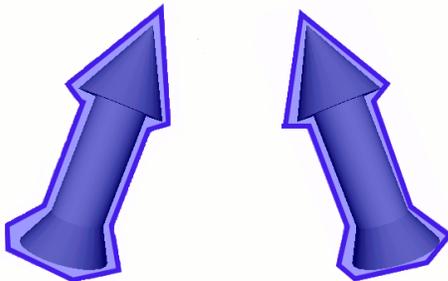


Figura 4 Modelo de Invólucro baseado em Politopo

### 3 ADAPTATIVIDADE EM DETECÇÃO DE COLISÃO

A adaptatividade proposta por [5] é aplicada no pipeline de detecção de colisão paralelo executado em arquitetura multi-core, permite dinamicamente adaptar a repartição de paralelismo. A seqüencialidade tradicional das duas principais fases (*broad* e *narrow*) é quebrada, conforme ilustra o processo paralelo da Figura 5. As fases são definidas como *threads* cuja configuração de repartição pode variar de acordo com a aplicação. Dois exemplos de configuração de repartição para aplicações diferentes:

- configuração 4-4: quatro *threads* da *Broad* e quatro da *Narrow*;
- configuração 5-1: cinco *threads* da *Broad* e uma da *Narrow*.

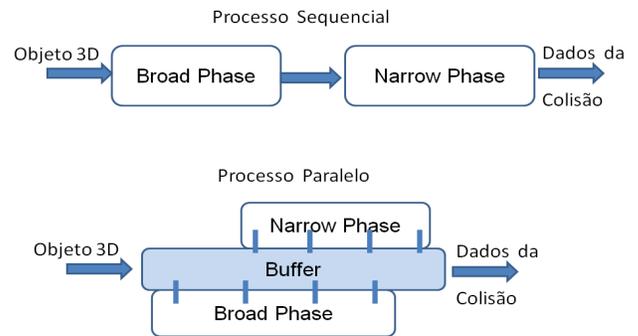


Figura 5 Processo Sequencial X Paralelo[5]

Aplica-se uma estrutura de *buffer* para compartilhar pares de objetos entre *threads*. Quando a *thread* da fase *Broad* identifica potencial colisão entre um par de objetos, por meio do algoritmo *Sweep and Prune*<sup>1</sup>, envia-os para o *buffer*. A *thread* da fase *Narrow* recebe esse par e verifica se há colisão utilizando o algoritmo apropriado para a forma geométrica, como o GJK(Gilbert-Johnson-Keerth) para os objetos complexos e o AABB para os objetos não convexos.[5]

Para distribuir as *threads* nas fases do *pipeline* e otimizar a performance do multi-core, [5] propõe uma técnica de balanceamento dinâmico de carga. No início da aplicação, por um curto período, é analisado o comportamento e a performance de cada fase. A configuração das *threads* varia de acordo com a demanda de tempo de cada fase.

A detecção de colisão para malhas de polígonos deformáveis[6]e[7] em simulações interativas, é outro algoritmo adaptativo. O algoritmo está baseado em adaptatividade de hierarquias de volumes envoltórios (*bounding volume hierarchies-BVH*).

<sup>1</sup> *Sweep and Prune* ou *Sort and Sweep* é um algoritmo que mantém os objetos 3D em alguma ordenação espacial, preferencialmente, em uma lista ligada duplamente encadeada, o que possibilita a atualização da lista com complexidade  $O(n)$ . [3]

Os tipos de modelos considerados nos algoritmos são: malhas de polígonos que são deformadas por reposicionamento arbitrário de vértices, modelos deformados por *morphing* linear com um número fixo de malhas referência e modelos de movimento completamente desestruturado relativo entre as primitivas geométricas[7].

O algoritmo proposto possui duas fases[6]:

1ª. *Broad* – fase na qual é utilizado o método *Sweep and Prune* para identificar corpos que estão próximos;

2ª. *Narrow* – utiliza árvore de volume envoltório para determinar a intersecção entre os corpos.

A árvore de volume envoltório é uma árvore binária construída em um estágio de pré-processamento, por repetidas divisões da geometria[6].

Para a construção da árvore, foram utilizadas duas estratégias.

A primeira estratégia foi baseada na divisão da forma inicial do corpo, dependendo do grau máximo de um nó de árvore, um pai AABB é dividido ao longo de um, dois ou três eixos de coordenadas em dois, quatro ou oito sub-volumes. O ponto médio de cada primitivo geométrico é atribuído a um desses volumes e um nó é criado para cada sub-volume não vazio. Para árvore de grau dois, o pai AABB é dividido ao longo de seu lado mais longo. Se é de grau quatro, a divisão é feita ao longo dos dois lados mais longos. E se é de grau oito, todos os três lados dos pais AABB são separados.[6]

A segunda estratégia calcula a média pelo ponto médio de todos os polígonos e os valores para a divisão do plano são escolhidos a partir desse ponto. [6]

Durante a execução, podem ocorrer mudanças na geometria dos objetos, ou novos objetos podem ser adicionados, portanto gera atualização da hierarquia.

A técnica de detecção de colisão de objetos proposta por [8] é destinada a ambientes distribuídos de realidade virtual possui a interferência da latência da rede.

Segundo [8], a posição e orientação de objetos nas máquinas conectadas à rede são enviadas por meio de mensagem ao servidor. O problema é que devido à latência, a mensagem somente chega ao servidor depois de algum tempo que foi enviada, assim esses dados já podem estar desatualizados. A predição adaptativa de colisão é uma técnica de detecção de colisão que identifica objetos em movimento com potencial colisão, calculando o tempo de colisão em função da distância e velocidade dos mesmos. Dois objetos estarão próximos se as esferas que os envolvem estão intersectadas ou se elas irão se intersectar em um certo tempo  $\Delta_T^2$ .

A abordagem do artigo procura compensar baixas taxas de atualização. Calcula-se o tempo de colisão para dois objetos em movimento. Se este tempo está abaixo de um limiar, os parâmetros para o protocolo de estimativa da posição atual são atualizados para permitir maior taxa de atualização, o limiar baseado em erro é reduzido ao menor valor.

Para dois objetos, verifica-se se suas esferas limitantes se intersectam no momento  $t=0$ . Se a verificação for verdadeira, nenhum outro cálculo é realizado, a intersecção foi

identificada. Caso as esferas limitantes ainda não tenham se intersectado, sua posição atual e velocidade são usadas para calcular o momento em que as esferas limitantes irão se intersectar.

Quando ocorre a colisão, é possível que dois computadores da rede tenham resultados diferentes. Essa diferença pode ocorrer porque o erro de aproximação pode ser maior para objetos com maior velocidade. Assim, é necessário de um processo, denominado resolução de decisão, para escolher qual computador é o responsável pela detecção de colisão. Um meio de tomar essa decisão é utilizar um número único de identificação para o computador, o computador com o menor número é o responsável por detectar as colisões e distribuir os resultados. Outro critério para definir qual computador é o responsável por detectar a colisão, é escolher aquele onde está o objeto com maior velocidade. E também, pode-se decidir com base no campo de visão para evitar inconsistências visuais.

Outra técnica que aplica a adaptividade é LBG que possibilita a divisão da superfície de objetos 3D em sub-áreas, e em cada uma há uma partícula. As partículas de um objeto 3D são classificadas em relação às partículas de outro objeto 3D candidato a colidir. Essa classificação é realizada pelo cálculo da força de atração entre as partículas, como ilustra a **Figura 6**. [2]

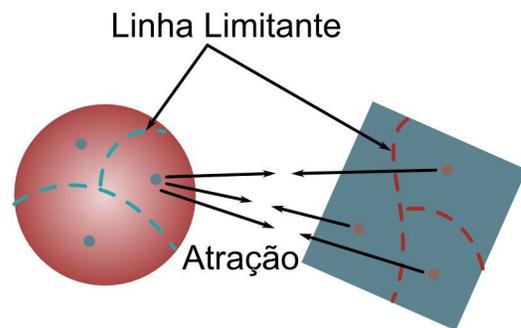


Figura 6 Interação de Força em uma Partícula baseado em [2]

O processo de partição do LBG consiste em determinar a quantidade de partículas usando a quantidade de objetos candidatos à colisão com este. Particionar a superfície, subdividindo-a em áreas, como ilustra a Figura 7. E em seguida, realizar a detecção de colisão, dividida em três partes:

- Cálculo da força de interação – é calculada a força aplicada a cada partícula, de acordo com a distância entre as partículas, quanto menor a distância maior será força de atração.
- Movimento das partículas – cada partícula movimenta-se apenas ao longo dos vértices na direção do vértice vizinho de onde recebeu força máxima de atração.
- Distância da colisão – encontrar o par de partículas mais próximas.

<sup>2</sup>  $\Delta_T$  é o limiar de proximidade.

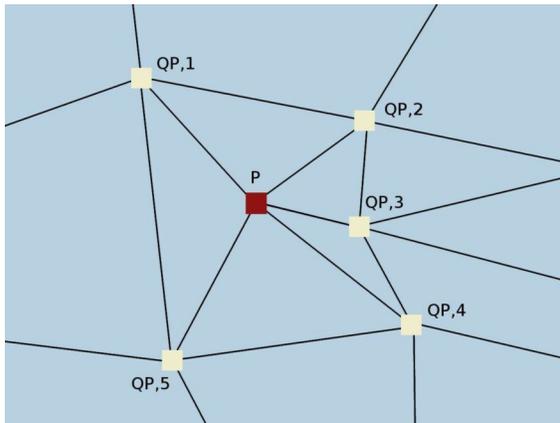


Figura 7 Posição da Partícula(P) e suas Coordenadas Vizinhas(QP) baseada em [2]

Quando ocorre grande deformação de um objeto, é realizado o reparticionamento, de modo que uma partícula não seja o suficiente para detectar a colisão em determinada área[2].

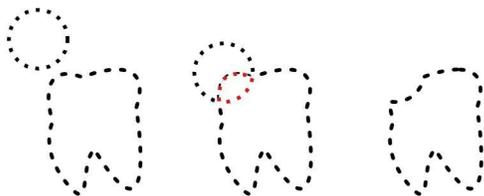
#### 4 SISTEMAS QUE UTILIZAM A ADAPTIVIDADE PARA DETECÇÃO DE COLISÃO

A busca pela aplicação de adaptatividade em detecção de colisão teve como parte do resultado dois sistemas[9]e[10]. Os dois utilizam com estrutura básica a Octree. O primeiro sistema apresentado por [9] usa modelos surfels<sup>3</sup>, ilustrado pela Figura 8, para renderizar as ferramentas do dentista e modelos de dentes virtuais para obter melhor qualidade visual com menos custo de memória.

O sistema apresentado por [9], baseado em surfels, utiliza a operação booleana para determinar quais pontos foram adicionados ou removidos do modelo. Esse sistema possui dois modos de uso. Um possibilita a escultura do modelo, o que deforma e gera sensação de força. Enquanto que o outro, não possibilita a deformação do modelo, e também não fornece a sensação de força.

Durante a interação, quando ocorre a colisão da broca com o dente, os pontos são removidos do modelo e a estrutura de dados atualizada. Como o modelo está representado por uma Octree, apenas o nó referente à parte removida é atualizado.

Ferramenta do dentista



Modelo do dente

Figura 8 Modelos usando Surfels baseado em [9]

<sup>3</sup> Surfel é a combinação de duas palavras surface (superfície) e element (elemento). [9]

Durante a busca por colisão na Octree, determina-se o nível que será calculado. O nível é determinado de acordo com a velocidade do movimento da broca, se é movimentada lentamente, a colisão é calculada com o menor nível da Octree. Se é movimentada rapidamente, será calculada com o nível 2.

#### 5 CONSIDERAÇÕES FINAIS

A revisão sistemática foi realizada na base do IEEE e da ACM. A chave de pesquisa foi ((adaptive collision detection) AND virtual reality). O resultado foi a indicação de dez artigos, porém apenas seis artigos foram selecionados para a pesquisa porque discutiam alguma técnica adaptativa.

Adaptatividade nas técnicas discutidas:

- configuração da repartição de acordo com a demanda nas fases Broad e Narrow[5];
- reconstrução da estrutura de dados depois da deformação do objeto 3D[6];
- repartição do objeto 3D após grande deformação[2];
- aplicação a adaptatividade na reação a um estímulo de colisão da broca com o dente, reorganizando o modelo de acordo com a velocidade do aparelho[9].

As próximas etapas na pesquisa é testar o desempenho e precisão dos algoritmos que utilizam a tecnologia adaptativa para aplicação futura em projetos na área de simulação para ensino na área de saúde e biológicas.

#### REFERÊNCIAS

- [1] R. M. S. Almeida. O cirurgião cardiovascular como intervencionista. Revista Brasileira de Cirurgia Vascular, 24(2), Suplemento: 35s-37s, 2009.
- [2] N. Saenghaengtham, P. Kanongchaiyos. Using LBG quantization for particle-based collision detection algorithm. *Journal of Zhejiang University SCIENCE A*, 7(7):p.1225-1232, CN, 2006.
- [3] C. Fünfzig, D.W.Fellner. Easy Realignment of k-DOP Bounding Volumes. Proceedings 9<sup>th</sup> Symp. Virtual Reality Software and Technology (VRST'02), pp.121-128, 2002.
- [4] C. Ericson. Real-time collision detection. Morgan Kaufmann, 2005.
- [5] Q. Avril, V. Gouranton and B. Arnaldi, Synchronization-free parallel collision detection pipeline. *20th International Conference on Artificial Reality and Telexistence*, p.22-28, Australia, 2010.
- [6] T. Larsson, T. Akenine-Möller, Collision Detection for Continuously Deforming Bodies. *Eurographics*, p.325-333, 2001.
- [7] T. Larsson, *Adaptive Bounding Volume Hierarchies for Efficient Collision Queries*. Suécia, Mälardalen University, 2009. 81p. Tese de Doutorado.
- [8] J. Ohlenburg. "Improving collision detection in distributed virtual environment by adaptive collision prediction tracking". *Proceedings of IEEE VR*, pp.83-90, Mar.2004.

- [9] H.T Yau, C.Y. Hsu. Development of a Dental Training System Based on Point-Based Models. Pp.779-787
- [10] E. I. Konukseven, M. E. Önder, E. Mumcuoglu, R. S. Kisnisci. Development of a Visio-Haptic Integrated Dental Training Simulation System. Journal of Dental Education. pp880-891, Ago.2010.