

Algoritmos Adaptativos em Detecção de Colisão em Aplicações para a Área de Saúde

A. Zotovici, R. Nakamura

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.

I. INTRODUÇÃO

ATUALMENTE, muitas aplicações utilizam a realidade virtual para simulação de elementos com os quais seria difícil ou impossível de interagir. Os elementos podem ser aqueles que existem no mundo real ou fictícios. Há muitas oportunidades de aplicação dessa tecnologia no ensino de diversas disciplinas.

Conforme [1], no ensino de Medicina, a simulação de procedimentos como cirurgia endovascular, que utiliza cadáveres humanos, proporciona alto grau de realidade. Porém não é um processo validado e usa apenas uma vez o material endovascular, o que gera alto custo.

Por outro lado, se é utilizada a realidade virtual, é possível proporcionar alto grau de realismo, simula-se a mudança das especificações do paciente 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 a tecnologia adaptativa sem a restrição de serem específicas para

objetos deformáveis. Após a identificação dessas técnicas, será estudado o seu algoritmo para futura aplicação em objetos deformáveis de sistemas de simulação educacional para a área de saúde.

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 utilizam técnicas adaptativas. 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.

II. DETECÇÃO DE COLISÃO

Esta seção discute algoritmos básicos de detecção de colisão com o objetivo de apresentar seu funcionamento, já que são empregados nas técnicas mais elaboradas de 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.

No caso de 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. Durante a detecção de colisão pode ocorrer a intersecção das caixas sem que os objetos tenham colidido porque esse volume envoltório não fica totalmente ajustado. A vantagem desse volume envoltório é que exige baixo consumo de memória e processamento rápido. Ele pode ser utilizado para verificar objetos que são candidatos à colisão.

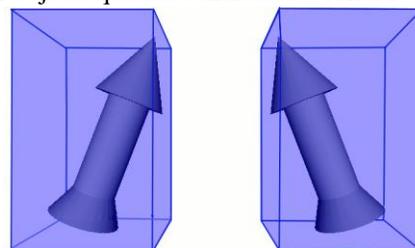


Figura 1 Caixas Alinhadas

Na abordagem utilizando caixas orientadas (Oriented Bounding Box-OBB), o volume envoltório possui a orientação e inclinação de acordo com o objeto que envolve, conforme ilustra a Figura 2. Essa transformação possibilita melhorar a precisão na identificação da intersecção. Porém ainda não é tão justo.

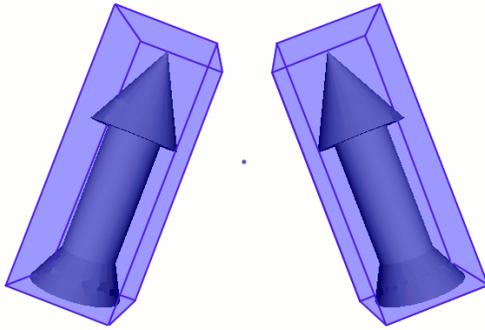


Figura 2 Caixas orientadas

No caso do uso de esferas, os objetos candidatos a colisão são envolvidos por uma esfera, conforme ilustra a Figura 3. Nesse algoritmo também pode ocorrer a intersecção do volume envoltório sem que os objetos tenham colidido. A vantagem do uso desse algoritmo é seu desempenho e pode ser usado para identificar os objetos candidatos à colisão.

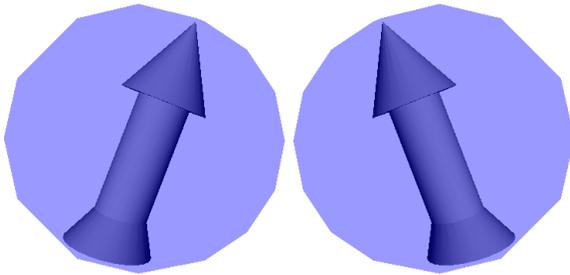


Figura 3 Esferas

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 $D6=(0,0,1)$ [3].

Os politopos podem criar um volume envoltório mais justo ao objeto, como ilustrado pela Figura 4. Mas exige mais memória e processamento que os volumes envoltórios discutidos no início da seção.

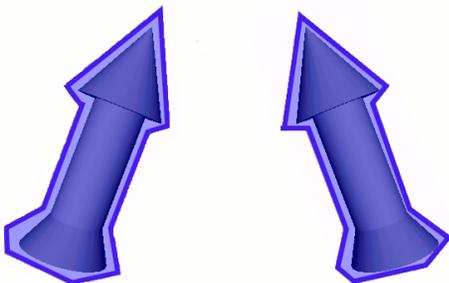


Figura 4 Modelo de Invólucro baseado em Politopo

Outra abordagem para detecção de colisão é a partição hierárquica do espaço que pode ser realizada por meio de uma *octree*.

O volume do espaço de um ambiente virtual tridimensional é dividido gerando uma árvore na qual cada nó pai tem oito nós filhos, o que gera uma divisão similar à Figura 5. Cada nó dessa árvore corresponde a uma partição do ambiente [4].

A divisão ocorre recursivamente até que um critério de parada seja atendido. O critério pode ser um grau máximo para a árvore ou tamanho mínimo para os cubos [4].

Esse processo possibilita mais precisão na detecção da colisão.

Para verificar a colisão, à medida que um objeto se movimenta no espaço, verifica-se a colisão quando há mais de um objeto no mesmo nó [4].

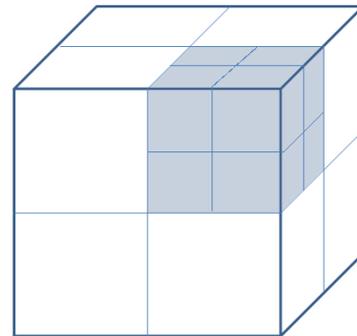


Figura 5 Partição de um cubo

III. METODOLOGIA DE REVISÃO

A metodologia de revisão adotada foi a revisão sistemática que, segundo [5], é uma metodologia rigorosa de pesquisa bibliográfica que difere da revisão de literatura comum pela definição do protocolo de pesquisa. O protocolo de pesquisa de pesquisa documental define critérios definidos durante a fase de planejamento: objetivos da revisão, questão de pesquisa, estratégia de pesquisa e recursos, idioma, palavras-chaves de busca, critério de inclusão de documento para estudo, critério de exclusão de documento, método de seleção de estudos.

Na fase seguinte, a condução da pesquisa, são registrados: a data de busca, a fonte, a palavra-chave utilizada, o título e autores de cada documento encontrado. Os documentos encontrados que atendem os critérios de inclusão são separados para a próxima fase, a análise de resultados.

Durante a análise de resultados, os dados do documento são registrados, realiza-se a síntese dos resultados e a conclusão sobre a pesquisa.

A revisão sistemática foi utilizada na pesquisa sobre técnicas e sistemas discutidos nas seções IV.

O objetivo dessa revisão foi realizar o levantamento da aplicação de técnicas adaptativas em detecção de colisão para sistemas de realidade virtual.

A questão desta revisão foi “Como são aplicadas as técnicas adaptativas em detecção de colisão nos sistemas de realidade virtual?”.

A estratégia para a pesquisa foi busca de artigos que discutam o estado da arte em técnicas adaptativas para detecção de colisão em ambientes que utilizam a realidade

virtual. Os recursos utilizados para a busca foram o Google Acadêmico, publicações do IEEE e ACM.

O idioma para as palavras-chave da busca foi o inglês. Esse é o idioma das publicações do IEEE e ACM por isso foi o escolhido.

As palavras-chave utilizadas foram: "Adaptive collision detection" "virtual reality".

A chave de pesquisa no Google Acadêmico foi “*adaptive collision detection*” com oito resultados, mas apenas quatro atendiam os critérios de inclusão. Um item do resultado foi uma tese, a partir da qual foram encontradas citações de artigo do mesmo autor que contribuíram no entendimento de seu trabalho. No IEEE, a chave de pesquisa foi ((*adaptive collision detection*) AND *virtual reality*) que retornou quatro artigos, dos quais apenas dois foram selecionados. E na ACM, a chave de pesquisa foi “*adaptive collision detection*” and “*virtual reality*”, para a qual não foi obtido resultado.

O critério de inclusão foi escolher artigos e livros que abordam algoritmos e exemplos de aplicação de técnicas adaptativas para detecção de colisão em realidade virtual. As publicações deveriam ter tempo inferior a cinco anos ou que discutisse conceitos fundamentais para detecção de colisão ou técnicas adaptativas que os artigos mais atuais não discutiram.

O critério de exclusão foi dispensar publicações com tempo superior a cinco anos, que não seja conceito fundamental para detecção de colisão e técnicas adaptativas, publicação não acadêmica, artigos em que a técnica adaptativa não é discutida por meio de algoritmos ou exemplos de sua aplicação.

A seleção de estudos foi baseada na seleção preliminar realizada por meio da leitura do *abstract* do artigo para identificar se atende aos critérios de inclusão, caso não fique explícito, o artigo será incluído. E a seleção final foi realizada por meio da leitura completa do artigo e o que possibilitou a definição sobre sua inclusão.

Durante a extração de dados, para cada artigo, foram documentados: dados para a referência do mesmo, objetivo do trabalho, descrição da técnica adaptativa aplicada à detecção de colisão, comentários relacionados à precisão e tempo de detecção de colisão, e conclusões.

A síntese de dados foi baseada na análise a aplicação de técnicas adaptativas em detecção de colisão e comparação das abordagens o que foi relatado na seção VI. .

IV. ADAPTATIVIDADE EM DETECÇÃO DE COLISÃO

Esta seção discute algoritmos de detecção de colisão que utilizam técnicas adaptativas com o objetivo de analisar como essas técnicas são aplicadas, isto é, se são aplicadas para adaptar o processo ou na reconstrução da estrutura de dados após alguma deformação.

A adaptatividade proposta por [6] é aplicada no *pipeline* de detecção de colisão paralelo executado em arquitetura multi-core. Essa arquitetura permite dinamicamente adaptar a repartição de paralelismo, eliminando a seqüencialidade tradicional das duas principais fases (*broad* e *narrow*), conforme ilustra o processo paralelo da Figura 6. As fases são definidas como *threads* cuja configuração de repartição pode variar de acordo com a necessidade da aplicação.

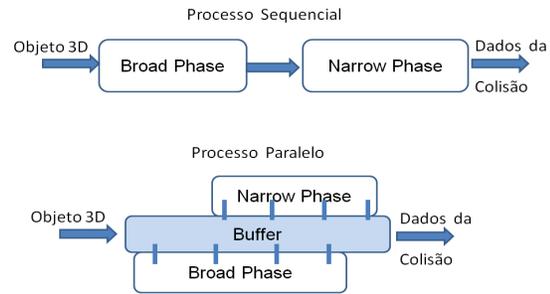


Figura 6 Processo Sequencial X Paralelo[6]

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*.

No processo paralelo, 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*¹, envia-os para o *buffer*. A *thread* da fase *Narrow* recebe esse par e, de acordo com a forma geométrica, utiliza o algoritmo apropriado para verificar se há colisão. Como por exemplo, o GJK(Gilbert-Johnson-Keerth) para os objetos complexos e o AABB para os objetos não convexos.[6]

Para distribuir as *threads* nas fases do *pipeline* e otimizar a performance do multi-core, [6] 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[7]e[8] em simulações interativas, é outro algoritmo adaptativo. O algoritmo está baseado em adaptatividade de hierarquias de volumes envoltórios (*bounding volume hierarchies-BVH*).

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[8].

O algoritmo proposto possui duas fases[7]:

- 1ª. *Broad* é a fase na qual se utiliza o método *Sweep and Prune* para identificar corpos que estão próximos;
- 2ª. *Narrow* é a fase que utiliza árvore de volume envoltório para determinar a intersecção entre os corpos.

¹ *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]

A árvore de volume envoltório é uma árvore binária construída em um estágio de pré-processamento, por repetidas divisões da geometria[7].

Para a construção da árvore, são utilizadas duas estratégias.

A primeira estratégia foi baseada na divisão da forma inicial do corpo, que depende do grau máximo de um nó de árvore, para que se defina a divisão de um nó pai. A divisão pode ocorrer 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 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.[7]

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. [7]

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

A técnica de detecção de colisão de objetos proposta por [9] é destinada a ambientes distribuídos de realidade virtual porque esses ambientes possuem a interferência da latência da rede.

Segundo [9], 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 Δ_T^2 .

A abordagem do artigo procura compensar as baixas taxas de atualização. No algoritmo, 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, assim 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 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 adaptatividade 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 7.[2]

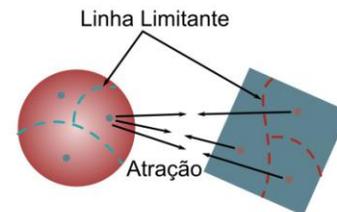


Figura 7 Interação de Força em uma Partícula baseado em [2]

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 8. E em seguida, realizar a detecção de colisão, dividida em três partes:

- Cálculo da força de interação é o processo no qual a força aplicada a cada partícula é calculada de acordo com a distância entre as partículas. A relação da força de interação é que, quanto menor a distância, maior será força de atração.
- Movimento das partículas é o processo em que cada partícula movimenta-se apenas ao longo dos vértices na direção do vértice vizinho de onde recebeu força máxima de atração.
- Distância da colisão consiste em encontrar o par de partículas mais próximas.

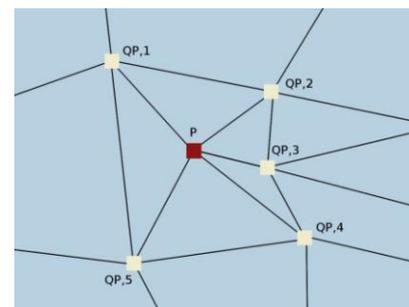


Figura 8 Posição da Partícula(P) e suas Coordenadas Vizinhas(QP) baseada em [2]

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].

² Δ_T é o limiar de proximidade.

V. SISTEMAS QUE UTILIZAM A ADAPTATIVIDADE PARA DETECÇÃO DE COLISÃO

Esta seção apresenta dois sistemas[10] que foram encontrados durante as buscas. Os dois utilizam como estrutura básica a *Octree*.

O sistema apresentado por [10] usa modelos surfels³, ilustrado pela Figura 9, para renderizar as ferramentas do dentista e modelos de dentes virtuais para obter melhor qualidade visual com menos custo de memória.

O modelo 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 9 Modelos usando Surfels baseado em [10]

A detecção de colisão utiliza tecnologia adaptativa porque durante a busca por colisão na *Octree*, o nível dessa estrutura hierárquica será calculado de acordo com a velocidade do dispositivo *haptic* que representa a broca. Se é movimentada lentamente, a colisão é calculada com o nível máximo da *Octree*. Se é movimentada rapidamente, será calculada com o nível dois.

A abordagem de detecção de colisão, proposta por [11], é para simulação de cirurgia intestinal. Nessa abordagem, pares de pontos próximos são rastreados o tempo todo. Durante este processo há partes do mesmo objeto, com estrutura tubular ou de membrana, colidindo devido ao formato do órgão. Para identificar essas colisões, utiliza o método de amostragem adaptativa para detecção da curvatura do intestino.

Para calcular a detecção da curvatura, encontra-se a curva *spline* por uma aproximação interpolando pontos de amostragem mecânica do intestino. Segundo [11], há dois passos para o cálculo da detecção: um para as extremidades do segmento e outro para as regiões de alta curvatura.

VI. CONSIDERAÇÕES FINAIS

Após a síntese dos dados obtidos com a revisão sistemática, foi possível verificar a aplicação da adaptatividade nas técnicas discutidas:

- configuração da repartição de acordo com a demanda nas fases *Broad* e *Narrow*[6];

- reconstrução da estrutura de dados depois da deformação do objeto 3D[7];
- repartição do objeto 3D após grande deformação[2];
- predição adaptativa de colisão[9] foi incluída no trabalho devido a sua abordagem para sistemas em rede;
- reação a um estímulo de colisão, como o da broca com o dente, reorganizando o modelo de acordo com a velocidade desse aparelho[10];
- detecção de colisão com diferentes partes o mesmo objeto[11].

A próxima etapa na pesquisa é testar e comparar 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. Fünzig, D.W.Fellner. Easy Realignment of k-DOP Bounding Volumes. Proceedings 9th Symp. Virtual Reality Software and Technology (VRST'02), pp.121-128, 2002.
- [4] C. Ericson. Real-time collision detection. Morgan Kaufmann, 2005.
- [5] V. M. Gonçalves, F. L. S. Nunes, M. E. Delamaro. Avaliações de Funções de Similaridade em Sistemas de CBIR: Uma Revisão Sistemática. 6^o Workshop de Visão Computacional, pp.199-204,2010.
- [6] 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.
- [7] T. Larsson, T. Akenine-Möller, Collision Detection for Continuously Deforming Bodies. *Eurographics*, p.325-333, 2001.
- [8] T. Larsson, *Adaptive Bounding Volume Hierarchies for Efficient Collision Queries*. Suécia, Mälardalen University, 2009. 81p. Tese de Doutorado.
- [9] J. Ohlenburg. "Improving collision detection in distributed virtual environment by adaptive collision prediction tracking". *Proceedings of IEEE VR*, pp.83-90, Mar.2004.
- [10] H.T Yau, C.Y. Hsu. "Development of a Dental Training System Based on Point-Based Models". *Comput-Aided Design Applications*, pp779-787, 2006.
- [11] L. Raghupathi, L. Grisoni, F. Faure, D. Marchal, M. P. Cani e C. Chaillou. "Na Intestinal Surgery Simulator: Real-Time Collision Processing e Visualization". *IEEE Transactions on Visualization and Computer Graphics*, pp.708-719, vol.10, no.6, 2004.



Andréa Zotovici é Bacharel em Ciência da Computação na Universidade São Judas Tadeu, São Paulo, São Paulo, Brasil, em 1995. Obteve o título de mestre em Engenharia Elétrica pela Universidade de São Paulo (USP), São Paulo, Brasil, em 2003. Atualmente é professora da Universidade São Judas Tadeu (USJT), Fundação Santo André e FATEC. Pesquisadora no Laboratório de Tecnologias Interativas (Interlab) da Escola Politécnica da USP.



Ricardo Nakamura possui graduação em Engenharia Mecânica - Automação e Sistemas pela Universidade de São Paulo (USP), mestrado e doutorado em Engenharia Elétrica também pela USP. Atualmente é Professor Doutor da Escola Politécnica da USP, no departamento de Engenharia de Computação e Sistemas Digitais. Suas principais áreas de interesse são jogo digitais, realidade aumentada e interação humano-computador.

³ Surfel é a combinação de duas palavras *surface* (superfície) e *element* (elemento). [10]