

Laboratório VISGRAF

Instituto de Matemática Pura e Aplicada

Caixas Orientadas Envolventes na Verificacao de Objetos

Luis A. Rivera, Paulo C. Carvalho, Luiz Velho

Technical Report TR-02-05 Relatório Técnico

May - 2002 - Maio

The contents of this report are the sole responsibility of the authors.
O conteúdo do presente relatório é de única responsabilidade dos autores.

Caixas Orientadas Envolventes na Verificação de Contatos

Luis A. Rivera[†], Paulo C. Carvalho[‡], Luiz Velho[‡]

[†] CCT - Laboratório de Matemáticas,
Universidade Estadual do Norte Fluminense - UENF,
Av. Alberto Lamengo, 2000, Campos dos Goytacazes, RJ, Brasil
`rivera@uenf.br`

[‡] IMPA - Instituto de Matemática Pura e Aplicada
Estrada Dona Castorina, 110, 22460 Rio de Janeiro, RJ, Brasil
`{pcezar,lvelho}@visgrafimpa.br`

Abstract

A determinação de pontos de interseção e contatos entre fronteiras de objetos não é uma tarefa simples. Uma forma de facilitar essa operação usa-se estruturas hierárquicas para issolar rapidamente os pedaços de segmentos em possível interseção, para finalmente se computar localmente os pontos de contato. Neste trabalho formula-se um novo método de representação em estrutura hierárquica de caixas orientadas envolventes adaptado e ajustadamente de segmentos de fronteiras de objetos definidos por curvas B-splines cúbicas fechadas com perturbações. Cada par de caixas adjacentes de um nível define uma outra caixa orientada. Cada caixa é computada de forma adaptada e ajustada aos segmentos de fronteira do objeto usando-se indicadores estatísticos de segunda ordem. O resultado computacional, em animações de objetos rígidos de geometria complexa em computação gráfica, mostra que o método formulado é eficiente comparado com outros métodos existentes.

Key words: Detecção de contatos, interseção de curvas, interferência entre objetos, árvore binária, caixas orientadas.

1 Introdução

A Simulação no computador de um pedaço do universo demanda a análise de uma serie de casos. Por exemplo, os objetos que se movimentam ou estão em repouso requerem, entre outros casos, a análise de contatos e interseções (interferências). Os algoritmos que detectam essas interferências são agrupados em duas abordagens: estruturadas e comparação direta. As estruturadas requerem de espaços adicionais para se definir estruturas hierárquicas

[Hubbard, 1995], [Gottschalk et al, 1996], mas otimiza o tempo na detecção de interferências. Na comparação direta, aproveitam-se as características geométricas do objeto para verificar suas proximidades [Hahn, 1988], [Moorre e Wilhelms, 1988], [Baraff, 1992], [Kamat, 1993], [Lin, 1994]; não usa espaços adicionais, mas em alguns casos as comparações geométricas são demoradas e impossíveis de serem usados em aplicações em tempo real.

Os atributos polinomiais dos objetos são fundamentais de detecção de interferências existentes. Assim, o algoritmo incremental [Lin, 1994] detecta interferências usando os espaços Voronoi definidos por vértices, arestas e faces dos objetos. Interferências por interseções baseadas nos métodos Clipping e projeções dos atributos poligonais dos objetos [Hahn, 1988], [Moorre e Wilhelms, 1988] e [Kamat, 1993]. A técnica da testemunha [Baraff, 1992] usa planos de separação definida pela combinação de faces, arestas e vértices dos objetos convexos. A árvore esfera [Hubbard, 1995] é a árvore de caixas orientadas [Gottschalk et al, 1996] definem estruturas hierárquicas cobrindo recursivamente segmentos poligonais do contorno do objeto. Todos esses métodos não suportam objetos de geometria complexa e com perturbações¹ não homogêneas nos seus contornos.

Um método híbrido das duas abordagens pode ser apropriado para a detecção de interferências entre objetos com perturbações nos seus contornos. Para este objetivo, na Seção 2 é formulada uma visão de uma arquitetura de animação enfatizando a parte de detecção de interferências. A Seção 3 aborda as estruturas hierárquicas, em particular a geração da árvore de caixas orientadas. Na Seção 4, formaliza-se a detecção de interferências dos objetos. Na Seção 5, apresentam-se os resultados e discussões da implementação do método formulado; e finaliza-se na Seção 6 com as conclusões respectivas.

2 Arquitetura geral de uma animação

Uma arquitetura geral de uma animação é composta por três segmentos: modelagem dos elementos da cena, simulação do movimento, e apresentações e controles.

2.1 Objetos de geometria complexa

Os objetos do universo têm geometria complexa e não aproximações poligonais; eles têm detalhes ou rugosidades irregulares em seus contornos.

Considera-se, para efeitos de validação do módulo formulado neste trabalho, objetos em 2D de contornos definidos por curvas B-splines cúbicas periódicas e contínuas [Rogers e Adams, 1990]. Um objeto pode ser definido por n pontos de controle, $C = \{c_i\}_{i=0,\dots,n}$, os que geram n segmentos da

¹Irregularidades do detalhe que definem as rugosidades do contorno

curva, $\{f_i\}_{i=0,\dots,n}$. O contorno do objeto é dado por $\cup_{i=0}^n f_i$, possivelmente, com rugosidade definido por uma distribuição aleatoria normal $N(0, \sigma_i)$ de ruído. A variância σ_i indica o grau de irregularidade dos detalhes no contorno do objeto. Na definição, cada segmento é definido geometricamente e, se tiver, os detalhes através de um intervalo de tolerância, reproduzidos estocasticamente. A Figura 1 mostra o segmento f_j com perturbações dentro de uma caixa.

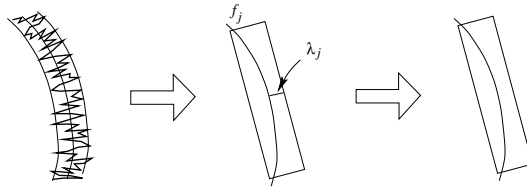


Figure 1: Um segmento do contorno com perturbações.

2.2 Simulação de movimento

Em um sistema de animação, o módulo de geração de movimento é executado em cada intervalo de tempo chamado “passo de tempo”. Ela, em geral, é baseada na dinâmica de Newton-Euler, que relaciona o movimento com as forças e torques aplicados sobre os corpos, através de umas equações diferenciais ordinárias (EDO). As equações são integradas numericamente para se obter os estados compostos por posições, orientações e velocidades dos objetos, com os quais são analisados situações de interferências, e, se houver contatos, são resolvidos a nível de velocidades dos objetos.

A Figura 2 mostra um trecho de um sistema de animação. Observa-se que com os novos estados temporários Y_1 , após cada passo de tempo, são verificadas as interferências por interpenetração. Se houver interpenetração, se estima um sub-passo dt , $0 < dt < dtt$, para convergir no contato, que logo é tratado. Os quadros de animação são apresentados em cada iteração em passo de tempo normal.

3 Estruturas hierárquicas

Uma estrutura hierárquica de envelopes permite descartar rapidamente as partes dos objetos que não estão em possível interferência. Um envelope deve conter um pedaço de contorno do objeto e seus detalhes, e permitir uma comparação rápida de interseções. As estruturas são árvores binárias, quaternárias ou octais; e os envelopes podem ser esferas, caixas retangulares isotéticas², elipses e caixas orientadas.

²Caixa de arestas paralelas ao sistema coordenado do universo

O envelope deve ser o mais ajustado e adaptado possível ao segmento envolvido, isto incrementa a possibilidade de detectar os segmentos em interferência. Por exemplo, uma esfera mínima cobrindo um segmento de contorno conteria mais espaço vazio do que uma elipse cobrindo o mesmo segmento; possivelmente, uma caixa orientada envolveria o segmento com pouco espaço vazio que a elipse e a esfera (vide Figura 3). Por outro lado, a comparação pela interseção entre duas esferas é mais rápida do que das caixas orientadas e elipses.

A árvore de caixas envolventes usadas em ray-tracing, cálculos de interseções e modelagem [Arvo e Kirk, 1989], [Rubin e Whitted, 1980], [Weghorst et al, 1984] é estendida por Gottschalk et al. [Gottschalk et al, 1996] para a detecção de interferências espaciais para objetos de contornos poligonais.

3.1 Caixas envolventes orientadas

Se cada segmento f_i e seus detalhes são envolvidos ajustadamente por uma caixa orientada b_i^0 , todo contorno do objeto estaria coberto por caixas mínimas chamadas *caixas básicas*. Em ordem de construir uma árvore binária de caixas orientadas, cada par de caixas básicas adjacentes, b_{2i}^0 e b_{2i+1}^0 , é coberta ajustadamente por uma outra caixa b_i^1 , chamada *caixa superior*. As caixas básicas de cada par de caixas adjacentes b_{2i}^1 e b_{2i+1}^1 são cobertas ajustadamente por uma outra caixa superior b_i^2 . Seguindo esse processo, constrói-se uma árvore binária, onde a raiz é uma super caixa envolvendo ajustadamente todas as caixas básicas do objeto. A inclusão de caixas básicas dentro da caixas superiores permite a conservação do segmento e seus respectivos detalhes. A Figura 4 mostra um segmento da árvore binária de caixas ori-

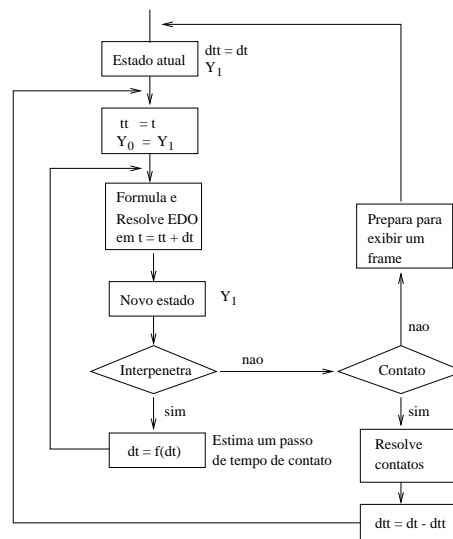


Figure 2: Fluxograma de simulação de movimento em uma animação.

entadas.

3.1.1 caixas básicas

A caixa básica b_i^0 é gerada a partir do segmento elementar f_i do contorno do objeto e suas perturbações em três fases: adaptação, ajuste e incremento. A adaptação está relacionada à orientação da caixa definida pelos eixos perpendiculares e_1 e e_2 de b_i^0 , que são os autovetores unitários da matriz de covariância σ_i determinada respeito à média simples μ dos m pontos p_i uniformemente amostrados sobre f_i . Esses dois autovetores são perpendiculares [Morrison, 1976]. Considerou-se $m = 5$ como adequado para computar a orientação dos eixos adaptados à tendência do segmento, já que maior concentração dos pontos em alguma parte de f_i pode desviar os eixos da tendência do segmento. Os elementos da matriz de covariância, neste caso,

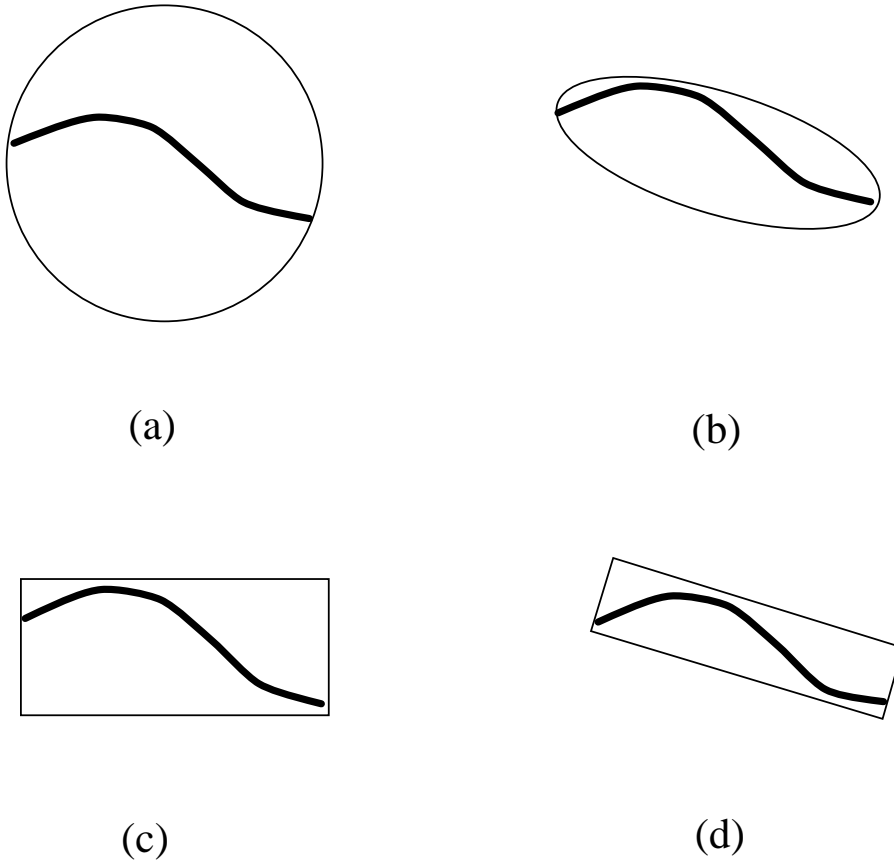


Figure 3: Tipos de envelopes: esfera, elipse, caixa isotética, caixa orientada.

são da forma:

$$\sigma_{xy} = \frac{1}{m-1} \sum_{i=1}^m (p_i^x - \mu^x)(p_i^y - \mu^y) \quad \text{com } \boldsymbol{\mu} = (\mu^x, \mu^y) = \frac{1}{m} \sum_{i=1}^m (p_i^x, p_i^y).$$

No ajuste, são projetados sobre os eixos, com origem em $\boldsymbol{\mu}$, os pontos máximos e mínimos, respeito aos eixos, do segmento f_i . Os lados da caixa são definidos pelos segmentos de maior dimensão entre as projeções sobre cada eixo.

Na fase de incremento, os lados da caixa são adicionados em uma tolerância λ_i que representa a cota, próximo ao superior, dos detalhes associados a f_i . O intervalo $\lambda_i = 0.9\sigma_i$ será um bom indicador para relacionar o intervalo de tolerância com a variância (σ_i) dos detalhes. Definidas as dimensões, recalcula-se o novo centróide da caixa b_i^0 . Cada caixa básica se constrói em tempo constante. As informações como os eixos unitários, os lados da caixa, o centróide da caixa, λ_i e o comprimento de f_i devem ser armazenadas na estrutura de dados que define a caixa orientada. Essas informações serão úteis nos cálculos de caixas superiores e nos testes pelas interferências. A Figura 5 mostra duas caixas básicas definidas por amostragem uniforme de dois tipos de segmentos de curva.

3.1.2 caixas superiores

As caixas formando nós intermediários e raiz da árvore binária são chamadas caixas superiores. Elas contêm duas ou mais caixas básicas; por isso, a

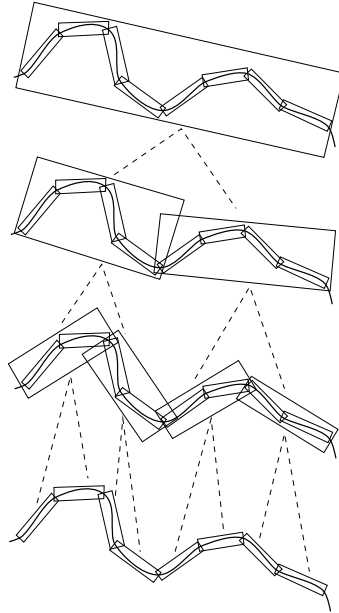


Figure 4: Hierarquia das caixas envolventes.

adaptação e ajuste são computados com outras considerações que as anteriores.

Propõe-se construir uma caixa superior B envolvendo adaptada e ajustadamente k caixas básicas adjacentes. Para isto, na adaptação, a matriz de covariância é computada com os centróides das caixas básicas ponderadas com o comprimento do segmento correspondente. Desta forma evita-se a influência de uma concentração de segmentos pequenos em um setor do contorno.

Se $\mathbf{p}_i = (p_i^x, p_i^y)$ é o centróide da caixa b_i^0 e l_i comprimento de arco do segmento básico f_i , a média de k segmentos é dada por

$$\boldsymbol{\mu} = \frac{1}{l} \sum_{i=1}^k l_i \mathbf{p}_i, \quad \text{com} \quad l = \sum_{i=1}^k l_i. \quad (1)$$

O elemento σ_{xy} da matriz de covariância $\boldsymbol{\sigma}$, é calculado, neste caso, como

$$\sigma_{xy} = \frac{1}{l} \sum_{i=1}^k l^i (p_i^x - \mu^x)(p_i^y - \mu^y). \quad (2)$$

No ajuste determinam-se as dimensões da caixa. Para isto, considera-se $\boldsymbol{\mu}$ origem da coordenada definida pelos eixos \mathbf{e}_1 e \mathbf{e}_2 (autovetores de $\boldsymbol{\sigma}$), sobre os quais são projetados os vértices das k caixas básicas. As distâncias dos pontos mais afastados de cada eixo são determinadas por

$$l_r^- = \min \{l_r^i\}_{i=1, \dots, 4k} \quad \text{e} \quad l_r^+ = \max \{l_r^i\}_{i=1, \dots, 4k}$$

onde $l_r^i = (\mathbf{v}_i - \boldsymbol{\mu}) \cdot \mathbf{e}_r$, $i = 1, \dots, 4k$ e $r = 1, 2$. Os quatro vértices de B são calculados combinando os quatro vetores que saem da origem $\boldsymbol{\mu}$, $l_1^+ \cdot \mathbf{e}_1$, $l_1^- \cdot \mathbf{e}_1$, $l_2^+ \cdot \mathbf{e}_2$ e $l_2^- \cdot \mathbf{e}_2$. Com esses elementos recalculam-se o centróide exato de B .

3.1.3 Avaliação da construção da árvore

Cada árvore é associada à estrutura de dados que define o objeto. Ela permanece inalterável durante o processo de animação. O número c de caixas da árvore para um objeto de $n = 2^k$ segmentos é dado por

$$c = n + \frac{n}{2} + \frac{n}{4} + \dots + \frac{n}{n} = 2n - 1.$$

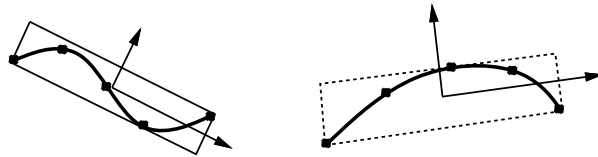


Figure 5: Eixos calculados por melhor ajuste com m pontos amostrados.

As folhas são compostas por n caixas, o nível anterior por $\frac{n}{2}$; em geral o nível j é formado por $n/2^j$ caixas. Cada caixa é construída em tempo constante, então a construção de uma árvore é feita em tempo linear, $O(n)$.

A Figura 6 mostra quadros do comportamento das construções das árvores respeito ao tempo. As medições foram feitas em uma estação *Sun SPARC-station 20* com *Solaris*, sendo o tempo tomado o mínimo dos tempos de várias repetições da cada caso, devido à política de trabalho da estação.

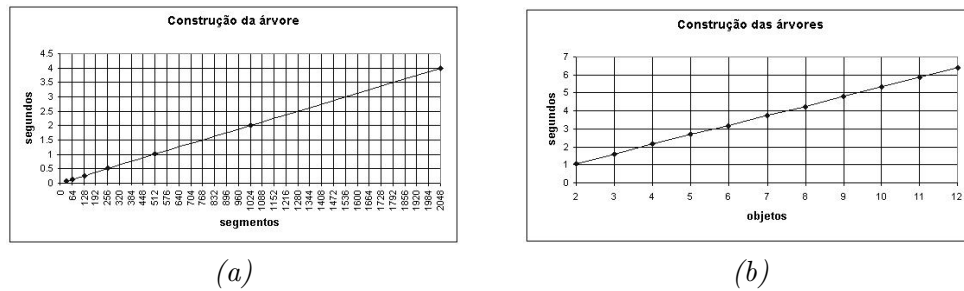


Figure 6: Tempo de construção de: (a) uma árvore variando número de segmentos; (b) árvores para objetos de 256 segmentos cada um.

4 Detecção de interferências

Se dois objetos se interceptam então algumas de suas caixas básicas se interceptam. A verificação de interferências é feita recursivamente começando pelas raiz das árvores através dos filhos enquanto for interseção das caixas. Ao finalizar, caso haja caixas básica registradas, significa que é possível que exista interseção dos dois objetos, a que será verificada analiticamente, ver detalhe em citekn:PrTeV+92. A Figura 7 mostra duas situações de dois objetos.

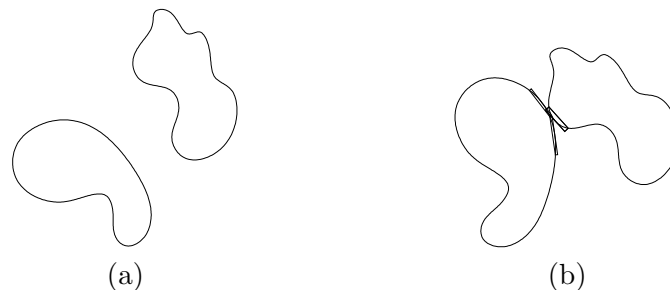


Figure 7: Situações: (a) separados; (b) em interferência.

Interseção de caixas

As caixas são fechos convexos, então duas caixas não se interceptam se existe uma reta que as separa (ver detalhes em [Rivera, 1996]). São projetados o par de caixas e seus centróides sobre uma reta paralela a um lado de uma das caixas (Vide Figura 8). Sejam A e B duas caixas, e \mathbf{L} a reta paralela a um lado da A. Os raios dos segmentos projetados, da caixa A, são obtidos como

$$r_A = \max\{r_A^j = |\mathbf{v}_A^j \cdot \mathbf{L}|, \text{ tal que } \mathbf{v}_A^j = (-1)^j \cdot l_A^1 \cdot \mathbf{e}_A^1 + l_A^2 \cdot \mathbf{e}_A^2, j = 1, 2\}, \quad (3)$$

onde l_A^i é a metade da longitude do lado correspondente a um eixo \mathbf{e}_A^i . O mesmo critério é utilizado no caso da caixa B.

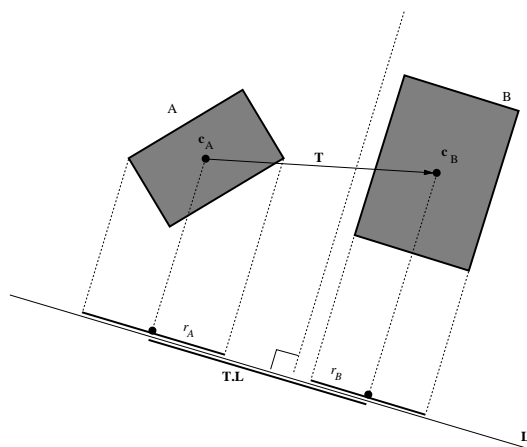


Figure 8: Projeção de caixas A e B sobre a reta L.

O raio r_A deve ser necessariamente a maior das projeções de \mathbf{v}_A^j (vetores que indicam os vértices da caixa a partir do centro) sobre \mathbf{L} , porque a caixa pode estar em uma situação em que nenhum de seus eixos sejam paralelos a \mathbf{L} , o que faria que a projeção de alguns desses vetores seja menor que a projeção real da caixa sobre \mathbf{L} . Se $\mathbf{T} = \mathbf{c}_B - \mathbf{c}_A$ o vetor que une os centróides das caixas A e B, a condição para que as caixas sejam disjuntas é

$$L > r_A + r_B, \quad \text{onde} \quad L = |\mathbf{T} \cdot \mathbf{L}|. \quad (4)$$

Como são quatro eixos que definem possíveis linhas de separação entre A e B, o teste da interseção envolve quatro casos. Se a condição (4) não for satisfeita em todos os casos, então as caixas A e B se intersectam. O primeiro indicador de não interseção é suficiente para considerar que não há interseção entre as caixas. Isto permite minimizar o número de operações pelas interseções, que no pior caso seria 4.

5 Resultados

O método proposto foi implementado em linguagem C e usou-se, para visualizações, as bibliotecas gráficas IUP/LED e CD desenvolvidas pelo grupo de tecnologia gráfica TecGraf PUC-Rio. O método é eficiente e robusto produzindo animações em tempo real.

Performance de detecção de interferências

A performance de OBBtree para objetos de contornos poligonais é discutida em [Gottschalk et al, 1996], mostrando eficiência superior a todos os outros métodos existentes. O método de herarquia de caixas envolventes para objetos complexos com perturbações herda de OBBtree a eficiência. Para ilustrar a performance do método, formula-se um exemplo de três situações de três objetos de 512 segmentos cada um. As situações são mostradas na Figura 9: objetos afastados, dois objetos em contato e, finalmente, uma situação em que existem múltiplos contatos entre todos os objetos. A execução de comparação das 3069 caixas orientadas distribuídas em três árvores foi efetuada na estação *Sun SPARCstation 20* com *Solaris*. O tempo medido é o mínimo tempo em segundos de vários experimentos devido ao sistema de multiprocessamento em que trabalha a máquina usada.

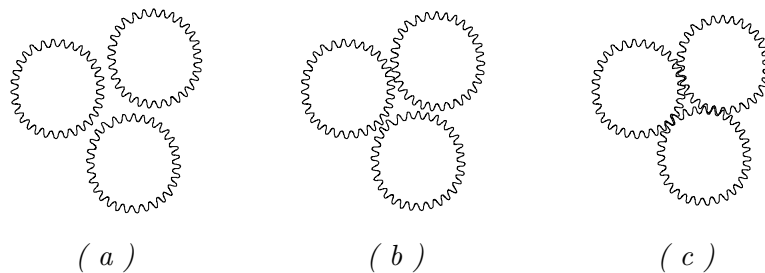


Figure 9: Objetos com 512 segmentos cada um: (a) afastados; (b) contato entre dois; (c) contatos entre todos.

A Figura 10 mostra uma tabela com os resultados obtidos. Observa-se que, mesmo quando há interpenetração dos objetos, apenas uma fração das caixas é comparada (459 das 3069) e o tempo de processamento é da ordem de 0,01 segundos.

Exemplos de resultados

Apresentam-se como resultados alguns frames de animações de objetos complexos. A Figura 11 mostra dois frames de duas cenas de animações com-

Número de objetos: 3		
Número total de segmentos: 1536		
Número total de caixas: 3069		
<i>situação</i>	<i>tempo em segundos</i>	<i>número de caixas comparadas</i>
(a)	0.0001	35
(b)	0.003	171
(c)	0.01	459

Figure 10: Resultados numéricos das situações da Figura 9.

postas por múltiplos objetos de geometria complexa com rugosidades variadas.

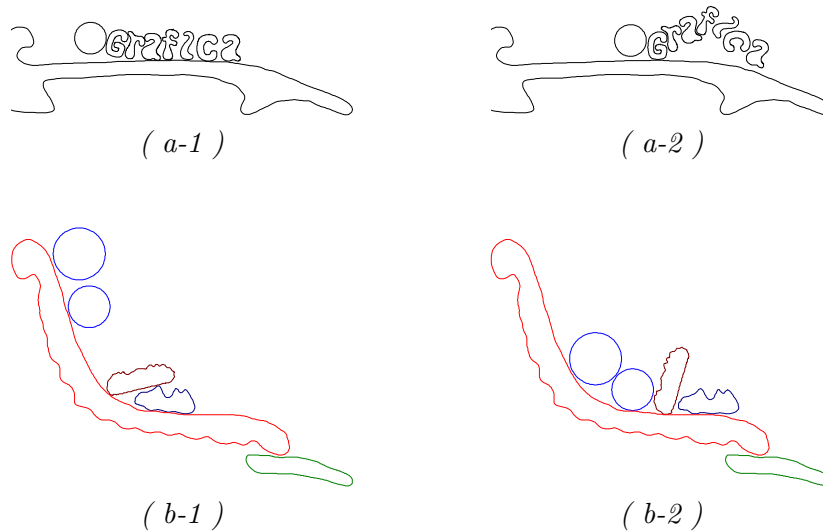


Figure 11: Dois frames de uma seqüência de duas cenas animações .

6 Conclusões e trabalhos futuros

Uma combinação adequada de estruturas hierárquicas procedimentos analíticos, na detecção de contatos, permite uma operação computacionalmente eficiente, em particular quando se envolvem objetos de geometria complexa. A comparação pela interseção de um par de caixas orientadas, segundo a estrutura de árvore binária, permite isolar rapidamente os pedaços do contorno que possivelmente estão em contato, que depois a verificação e os cálculos de pontos de contatos são feitos por procedimentos locais.

Uma aplicação do modelo formulado aqui é no problema de “cortes” de peças nas manufaturas. Cada objeto represente uma peça que se deseja cortar de um material inteiro, que pode ser couro para sapatos, tecidos, lâminas metálicas, cartões, etc. Os objetos se compactam seguindo as leis físicas, adicionando alguma heurística para permitir movimentar os objetos até uma situação estática.

References

- [Arvo e Kirk, 1989] J. Arvo e D. Kirk. A Survey of Ray Tracing Acceleration Techniques. *Introduction to Ray Tracing*, 1989, 201-262.
- [Baraff, 1992] D. Baraff. Dynamic Simulation of Non-penetrating Rigid Bodies. *PhD. Thesis*, Department of Computer Science, Cornell University, Ithaca, 1992.
- [Gottschalk et al, 1996] S. Gottschalk, M. Lin e D. Manocha. OBBtree: A Hierarchical Structure for Rapid Interference Detection. *Computer Graphics Proceeding, SIGGRAPH'96*, 1996, 171-180.
- [Hahn, 1988] J. Hahn. Realistic Animation of Rigid Bodies. *Computer Graphics*, v. 24, n. 4, 1996, 299-308.
- [Hubbard, 1995] P. Hubbard. Collision Detection for Interactive Graphics Application. *PhD. Thesis*, Department of Computer Science, Brown University, Berkeley, 1995.
- [Kamat, 1993] V.V. Kamat. A Survey of Techniques for Simulation of Dynamic Collision Detection and Response. *Computer & Graphics*, v.17, n.4, 1993, 379-385.
- [Lin, 1994] M. Lin. Efficient Collision Detection for Animation and Robotics. *PhD. Thesis*, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 1993.
- [Moorre e Wilhelms, 1988] M. Moore e J. Wilhelms. An Collision Detection and Response for Computer Animation. *ACM Computer Graphics*, v.22, n.4, 1988, 289-298.
- [Morrison, 1976] D. F. Morrison. Multivariate statistical methods. *McGraw-Hill series in probability and statistics*, 2 ed., 1976, 415 ps.
- [Press et al, 1992] W.H. Press, S.A. Teukolsky, W.T. Vetterling e B.P. Flannery. Numerical Recipes in C: The Art of Scientific Computing. *Cambridge*, 2 ed., 1992, 994 ps.

- [Rivera, 1996] L. Rivera. Simulação Dinâmica de Corpos Rígidos com Restrições de não Interpenetração. *Tese de mestrado*, Departamento de Informática, Pontifícia Universidade Católica (PUC-Rio), Rio de Janeiro, Brasil, 1996.
- [Rogers e Adams, 1990] D. Rogers e J. Adams. Mathematical Elements for Computer Graphics. *McGraw-Hill International Editions*, 2da ed., 1990, 611 pgs.
- [Rubin e Whitted, 1980] S. Rubin e T. Whitted. A 3-Dimensional Representation for Fast Rendering of Complex Scenes. *Computer Graphics Proceeding, SIGGRAPH'80*, 1980, 110-116.
- [Weghorst et al, 1984] H. Weghorst, G. Hooper e D. Greenberg. Improved Computational Methods for Ray Tracing. *ACM Transactions on Graphics*, 1984, 52-69.