

Luis Antonio Rivera Escriba

**Animação Baseada em Física com Modelos
Geométricos em Multi-Resolução**

Tese de Doutorado

Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro

Rio de Janeiro, 29 de setembro de 2000.

Luis Antonio Rivera Escriba

Animação Baseada em Física com Modelos Geométricos em Multi-Resolução

Tese apresentada ao Departamento de Informática da PUC-Rio, como parte dos requisitos para a obtenção do título de Doutor em Ciências da Informática.

Orientador: Prof. Paulo Cezar Pinto Carvalho

Co-orientador: Prof. Luiz Velho

Departamento de Informática

Pontifícia Universidade Católica do Rio de Janeiro

Rio de Janeiro, 29 de setembro de 2000.

© Luis Antonio Rivera Escriba, 2001.
Todos os direitos reservados.

*“A minha mãe Graciela
e a minhas irmãs Ada e Vilma pelo amor e in-
centivo prestados incondicionalmente ao longo
de toda minha vida.”*

Agradecimentos

Ao professor Paulo Cezar Pinto Carvalho, meu orientador, por suas acertadas sugestões, por estar sempre disposto a discutir todos os problemas, por suas críticas e observações constructivas, e por sua paciência na correção do texto.

Ao professor Luiz Velho, meu co-orientador, que em muitas ocasiões dedicou horas e horas de seu tempo para ouvir minhas propostas de pesquisa, e por suas valiosas sugestões no desenvolvimento desta tese.

Ao professor Marcelo Gattass pelo apoio a respeito do vínculo burocrático com a PUC-Rio, e por me permitir usar os recursos computacionais do TecGraf.

Ao professor Ruy Milidiú que me introduziu na teoria de *wavelets*.

Aos meus amigos e colegas Fábio Porto, Álvaro Barbosa, Fausto Ayres, Nestor Mamani, Cássia Baruque, Lúcia Baruque, Fernanda Lima e Anselmo Montenegro que, de uma ou outra forma, me apoiaram moralmente durante todo este período.

À Mariela Cortes por ler e corrigir uma versão deste trabalho.

À Isabela Alencar pela boa vontade no atendimento das respectivas tarefas administrativas.

Ao amigo Presbítero Zamora por seu apoio incondicional nos momentos críticos.

Tanto à PUC-Rio como ao IMPA pela oportunidade, e pelos novos conhecimentos adquiridos.

Em especial a minha mãe e irmãs que me brindaram carinho e apoio moral para continuar no curso de doutorado.

Finalmente, agradeço ao CNPq pelo apoio financeiro.

Resumo

O uso de leis físicas na geração de animações permite produzir efeitos reais de movimentos e tratamentos automáticos de colisões. Enfrenta-se, no entanto, dificuldades para executar eficientemente os cálculos numéricos necessários e para produzir efeitos interessantes, devido à complexidade da geometria dos objetos em animação. Nesta tese, formula-se um modelo para gerar animações baseados em modelos físicos de objetos de geometria arbitrária. Os detalhes irregulares das superfícies dos objetos são incorporados no cálculo da dinâmica de contatos dos objetos em colisão. Usa-se a teoria de multiresolução baseada na transformada wavelet para se relacionar os detalhes de superfícies do objeto, com estimadores estatísticos, ao grau de rugosidade das superfícies em referência. Esta abordagem também facilita a construção hierárquica de envelopes adaptados retangulares orientados (MOBBtree - *Multiresolution Oriented Bounding Box tree*), que preservam os detalhes das superfícies, para uma rápida detecção de possíveis segmentos de superfícies em colisão, que posteriormente são utilizados para determinar analiticamente os pontos de contatos tolerados. A dinâmica de contatos usa o paradigma baseado em impulsos que, junto com o grau de rugosidade local do ponto de colisão, fornece animações interessantes e visualmente realistas.

Abstract

The use of physical laws in the generation of animations permits the production of (ou aims at producing) real effects of movements and automatic treatment of collisions. However, a number of difficulties are encountered for the efficient execution of the needed numeric computations and for the production of interesting effects, due to the geometric complexity of the objects in the animation. In this thesis, we formulate a model to generate animations based on physical models of objects with arbitrary geometry. The irregular details of the object surfaces are incorporated in the calculation of the dynamics of the object contacts which are found to be in collision. We use multiresolution theory based on the wavelet transformation to establish a relationship between the details of the object surfaces and the surfaces rugosity degree, using statistical estimators. This approach leads to the hierarchical construction of oriented-rectangular adapted envelopes (MOBBtree - *Multiresolution Oriented Bounding Box tree*). They preserve the surfaces details, allowing for a fast detection of possible surface segments in collision, that are later used to determine the tolerated contact points. The contact dynamics uses the impulse-based paradigm that, simultaneously with the rugosity degree of the local collision point, generate interesting and visually realistic animations.

Conteúdo

Agradecimentos	v
Resumo	vi
Abstract	vii
1 Introdução	1
1.1 Motivação	2
1.2 Abordagens atuais e suas limitações	3
1.2.1 Detecção de interferências	4
1.2.2 Resolução de interferências	6
1.3 Objeto de geometria complexa	7
1.4 Proposta de pesquisa	8
1.5 Contribuições	9
1.6 Organização	10
2 Objetos de geometria complexa em multiresolução	12
2.1 Funções-base B-splines cúbicas periódicas contínuas	13
2.1.1 Curvas fechadas	14
2.1.2 Derivadas das funções B-Splines periódicas	16

2.2	Representação em multiresolução	17
2.2.1	Decomposição e composição do objeto em várias resoluções	18
2.2.2	Relação de coeficientes e filtros	20
2.2.3	Multiresolução Biortogonal	22
2.2.4	Pontos de controle do objeto em multiresolução	24
2.2.5	Refinamento dos pontos de controle do objeto	27
2.3	Alteração de forma do objeto	28
2.3.1	Manipulação direta do contorno	28
2.3.2	Deformação do objeto em múltiplas resoluções	30
2.4	Perturbações no contorno	34
3	Construção de estruturas hierárquicas	38
3.1	Estrutura hierárquicas	39
3.1.1	Estruturas MOBBtree	39
3.2	Determinação da caixa envolvente adaptada	42
3.2.1	Uso da matriz de covariância	43
3.2.2	Determinação dos eixos das caixas envolventes	44
3.3	Caixa básica	45
3.3.1	Determinação dos eixos da caixa envolvendo um segmento de curva	45
3.3.2	Caixa ajustada a um segmento de curva	48
3.4	Caixa superior a partir das caixas básicas	50
3.5	Árvore binária de caixas envolventes	52
4	Análise de perturbações	54

4.1	Detalhes do contorno em multiresolução	55
4.1.1	Perturbações de um contorno	58
4.2	Incorporação de perturbações nas caixas envolventes	61
4.3	Avaliação de construção de MOBBtree	64
5	Relação entre as perturbações e as tolerâncias	69
5.1	Intervalos de tolerância	70
5.1.1	Enfoque estatístico de intervalo de tolerância	71
5.2	Tolerâncias de segmentos de contorno refinado	73
5.3	Tolerância em multiresolução	75
5.3.1	Tolerâncias definidas por coeficientes wavelets	76
5.3.2	Tolerâncias definidas pelo contorno de detalhe	78
5.3.3	Tolerâncias atribuídas aos segmentos de resoluções inferiores	80
5.3.4	Limitações de contornos de referências	80
5.4	Tolerância relativa a um ponto	81
5.5	Atrito adotado do grau de intervalo de tolerância	83
5.5.1	Estimação do coeficiente de atrito	85
6	Movimento dos corpos com interferências	86
6.1	Dinâmica de movimento dos corpos	87
6.1.1	Estado de corpo em movimento	87
6.1.2	Evolução síncrona de movimento	89
6.2	Detecção de interferências	90
6.2.1	Detecção de interferências de alto nível	91
6.2.2	Interferências em hierarquía	95

6.2.3	Interferências em detalhe	99
6.2.4	Detecção de pontos de contato	102
6.3	Performance de detecção de interferências	107
7	Tratamento de pontos de contatos	110
7.1	Pontos de contato	112
7.2	Impacto baseado em impulso	113
7.2.1	Impulso em um ponto de contato	117
7.3	Contato contínuo	120
7.4	Exemplos de resultados obtidos de simulação dinâmica	123
8	Conclusões e trabalhos futuros	133
8.1	Conclusões	133
8.2	Trabalhos futuros	136
A	Tabela de filtros	139
	Bibliografia	142

Lista de Figuras

2.1	<i>Funções-base B-splines cúbica periódicas.</i>	14
2.2	<i>Uma curva B-spline cúbica com oito pontos de controle.</i>	15
2.3	<i>Segmentos de funções bases B-spline cúbica reparametrizadas para intervalo $[0, 1)$.</i>	16
2.4	<i>Um objeto complexo limitado por uma curva fechada B-spline cúbica com 32 pontos de controle.</i>	17
2.5	<i>Estrutura de transformação direta e inversa: (a) decomposição; (b) composição.</i>	19
2.6	<i>Um objeto com n pontos de controle, tratados em ciclo.</i>	24
2.7	<i>Decomposição de um objeto em dois níveis: (a) objeto original; (b) objeto decomposto no primeiro nível; (c) objeto decomposto no segundo nível.</i>	26
2.8	<i>Um objeto inicial de 32 pontos de controle refinado para 64 pontos.</i>	27
2.9	<i>Uma curva fechada B-spline cúbica deformada em três pontos de diferentes segmentos da curva.</i>	29
2.10	<i>Vetor de deslocamento $\Delta \mathbf{d}$ aplicado em um ponto de cada resolução da curva, $f^j(t_j)$, $3 \leq j \leq 6$, por vez (curva pontilhada externa é resultado depois de deslocar um ponto $f^3(t_3)$; curva de linha tracejada interna é resultado após o deslocamento do ponto $f^6(t)$).</i>	33

2.11	<i>Vetor de deslocamento $\Delta \mathbf{d}$ segmentado aplicado sobre curvas de resolução inferior. A curva de linha tracejada é produzida depois da aplicação dos segmentos de vetores, $\Delta \mathbf{d}_1, \Delta \mathbf{d}_2, \dots, \Delta \mathbf{d}_m$, sobre cada curva de resolução inferior f^6, f^5, f^4 e f^3, respectivamente.</i>	34
2.12	<i>Criação das perturbações utilizando distribuição de coeficientes wavelets em dois níveis.</i>	36
2.13	<i>Criação das perturbações utilizando deformações locais em dois níveis.</i>	37
3.1	<i>Estrutura de um segmento de OBBtree.</i>	40
3.2	<i>Hierarquia das caixas envolventes conservativas representadas em uma árvore binária.</i>	42
3.3	<i>Casos de segmentos de curva f_j e seus pontos de controle respectivos.</i>	46
3.4	<i>Caixas envolvendo um segmento de curva f_j: caixa de linha pontilhada relativo ao eixo determinado por extremos do segmento; caixa linha contínua relativo ao eixo determinado com melhor ajuste.</i>	47
3.5	<i>Eixo calculado por melhor ajuste com n pontos amostrados.</i>	48
3.6	<i>Caixas orientadas hierárquicas envolvendo segmentos do contorno de um objeto de 64 pontos de controle.</i>	53
4.1	<i>Relação dos elementos de curvas f^{j+1} e \bar{f}^{j+1}.</i>	58
4.2	<i>Perturbações de contorno de um objeto: (a) sobreposição de duas resoluções adjacentes de um contorno; (b) função de perturbação do contorno de menor resolução.</i>	59
4.3	<i>Definição de um sistema de coordenadas para \bar{d}_r^{j+1}.</i>	60
4.4	<i>Caixa aumentada envolvendo segmento f_r^{n-3} e suas perturbações refinadas relativas.</i>	62
4.5	<i>Construção de MOBBtree em segundos: (a) um objeto com número de segmentos variável; (b) objetos de 256 segmentos cada um.</i>	66
4.6	<i>Geração de caixas envolventes de segmentos de curvas em multiresolução.</i>	68

5.1	<i>Situações entre dois objetos: (a) contato entre A e B; (c) A e B interpenetrados; (b) A e B separados.</i>	70
5.2	<i>Vista ampliada de uma seção de aço finamente polida. Esta imagem aparece no texto <i>Physics, Holliday & Resnick, 1967.</i></i>	71
5.3	<i>Casos possíveis para o engrossamento de um segmento de contorno: (a) sem perturbação; (b) perturbação quase constante; (c) perturbação variada.</i>	72
5.4	<i>Exemplo de biblioteca de quatro tipos de tolerâncias.</i>	74
5.5	<i>Exemplo de um tipo de distribuição de tolerâncias da biblioteca em um contorno de 32 e 128 segmentos, respectivamente.</i>	75
5.6	<i>Exemplo de distribuição de tolerâncias tipo_3, com uma função de distribuição, sobre um contorno de 32 segmentos.</i>	76
5.7	<i>Dois objetos definidos: (a) sem mostrar detalhe de aspereça de seus contornos; (b) mostrando aspereças de seus contornos segundo a biblioteca de tolerâncias.</i>	76
5.8	<i>Objeto mais refinado define perturbações de sua representação menos refinada: (a) versão refinada; (b) versão menos refinada; (c) contorno de detalhe mapeado horizontalmente.</i>	77
5.9	<i>Seqüência de tolerâncias definidas pelos coeficientes wavelets do objeto da Figura 5.8: (a) primeiro nível grosseiro; (b) segundo nível grosseiro.</i>	78
5.10	<i>Seqüência de tolerâncias definidas pelo contorno de detalhe do objeto da Figura 5.8: (a) primeiro nível grosseiro; (b) segundo nível grosseiro.</i>	79
5.11	<i>Definição da tolerância do nível j: (a) tolerância relativa à resolução j + 1; (b) tolerância definida pela perturbação de j; (c) tolerância considerando todos as perturbações superiores.</i>	81
5.12	<i>Segmento da função de tolerância: (a) pontos de contato que ocorrem na região preenchida para $t_c \geq 0.5$; (b) pontos de contato que ocorrem na região preenchida para $t_c < 0.5$; (c) segmento da função de tolerância com parâmetro $1 \leq t_\lambda < 2$.</i>	84

6.1	<i>Projeção de caixas, envolvendo o objeto, sobre os eixos coordenados. .</i>	92
6.2	<i>Projeção de caixas A e B sobre a reta L: comprimento T.L maior que a semi soma dos segmentos projetados indica que as caixas não intersectam.</i>	96
6.3	<i>Estados de dois objetos na cena de animação: (a) momento inicial; (b) momento em que se detectam interferências entre as primeiras caixas básicas.</i>	100
6.4	<i>Trecho de código contendo lógica de detecção de contatos em detalhe.</i>	101
6.5	<i>Intervalo de domínio de cada segmento é dividido em quatro sub-intervalos.</i>	107
6.6	<i>Situação de três objetos com 512 segmentos cada um: (a) afastados; (b) contato entre dois objetos; (c) contato entre todos os objetos; (d) maior número de contatos entre todos os objetos.</i>	108
6.7	<i>Resultados da execução de quatro situações de três objetos de 512 segmentos cada um, segundo a Figura 6.6.</i>	109
7.1	<i>Colisão entre A e B, indicando alguns elementos que são considerados na análise de colisão. O eixo vertical do sistema de referência de colisão é alinhado com o vetor normal.</i>	116
7.2	<i>Um ponto de contato com o grau de penetração δ no intervalo de tolerância λ.</i>	121
7.3	<i>A função que define coeficiente de restituição para dinâmica de contato contínua. O valor de coeficiente de restituição depende do grau de penetração no intervalo de tolerância λ.</i>	122
7.4	<i>Um exemplo de dinâmica de contato para um objeto esférico que se movimenta sobre um objeto horizontal:(a) estado inicial; (b) estado em animação.</i>	123
7.5	<i>Comportamento do atrito em evento de colisão do corpo esférico em movimento sobre a superfície horizontal em repouso, com tipos de rugosidades variadas, para esfera - bloco horizontal: (a) tipo_1 - tipo_1; (b) tipo_1 - tipo_2; (c) tipo_1 - tipo_3; (d) tipo_2 - tipo_2; (e) tipo_2 - tipo_3; (f) tipo_3 - tipo_3.</i>	124

7.6	<i>Evolução da velocidade da bola rolando sobre uma superfície horizontal: (a) velocidade com atrito mostrado na Figura 7.5(a); (b) velocidade com atrito mostrado na Figura 7.5(e)</i>	125
7.7	<i>Um corpo de geometria irregular sendo deslizado sobre uma superfície horizontal estática.</i>	126
7.8	<i>Comportamento da velocidade relativa tangencial de contato dos corpos da Figura 7.7: (a) quando ambos os corpos têm rugosidades variadas; (b) quando o corpo móvel tem rugosidade nula e o horizontal fixo tem rugosidade variada (rugosa, lisa e rugosa).</i>	126
7.9	<i>Simulação do comportamento de quatro objetos em movimento sobre um bloco fixo de superfície irregular. Cada corpo tem rugosidades variadas nas suas superfícies refinadas e diferentes coeficientes de restituição ao contato.</i>	127
7.10	<i>Simulação do comportamento de múltiplos objetos de superfícies refinadas irregulares com rugosidades tipo_2.</i>	129
7.11	<i>Simulação do comportamento de múltiplos objetos de superfícies refinadas irregulares com rugosidades tipo_3.</i>	130
7.12	<i>Um exemplo de três objetos de geometrias arbitrárias em três resoluções.</i>	131
7.13	<i>Seqüência de quadros gerados pela animação de três objetos rugosos definidos por sua geometria de primeiro e segundo nível de resolução: coluna b definida por resolução de primeiro nível (-1); coluna c definida por resolução de segundo nível (-2).</i>	132

Capítulo 1

Introdução

As simulações dinâmicas de corpos livres ou com restrições têm sido extensivamente estudadas na engenharia, na robótica e na computação gráfica. Na animação por computador, a preocupação principal é que os movimentos dos objetos sejam perceptualmente semelhantes aos ocorridos no mundo real; para isto, são vários os fatores que devem ser considerados. Esses fatores podem ser reunidos em dois grandes grupos: os fatores de descrição dos objetos e os fatores de comportamento dos objetos.

O grupo de descrição refere-se à definição dos fatores que estabelecem a cena da animação, considerando as características principais dos objetos, tais como sua geometria e detalhes de sua superfície. Pode-se incluir neste primeiro grupo as características do material, como sua composição, e sua textura, entre outros. Em geral, este primeiro grupo inclui a modelagem e propriedades fundamentais dos objetos que representam o mundo real.

O grupo de comportamento inclui os movimentos atribuídos aos objetos e os efeitos adquiridos como consequência da interação entre eles. Estes comportamentos determinam as situações em que os objetos são mostrados em cada passo de tempo, constituindo uma série de *frames*.

Modelar comportamentos perceptivelmente coerentes com o mundo real é ainda objeto de muitas pesquisas, embora sucessos tenham sido alcançados nesta linha com a ajuda de modelos físicos e Métodos de Elementos Finitos. Há também um interesse especial a respeito de métodos que permitam a geração destes movimentos eficientemente, em tempo real. Em geral, a exatidão e a eficiência da análise de com-

portamento dos objetos, em computação gráfica, para fins de animação, são objetivos antagônicos. Os Métodos dos Elementos Finitos e abordagens similares são bons em exatidão, mas em eficiência não são bem recomendados para este tipo de animação. A tendência nas aplicações na Computação Gráfica é que os algoritmos sejam eficientes e a exatidão seja suficientemente aceitável.

A possibilidade de geração da animação em tempo real, com critérios de eficiência e exatidão aceitáveis, depende da complexidade dos objetos da cena (quantidade de objetos e de suas características geométricas). A animação envolvendo um grande número de corpos simples tem sido bastante estudada; como conseqüência, existem bons algoritmos em termos de eficiência. Pouca atenção tem sido dada ao caso em que os objetos possuem geometria complexa, com perturbações¹ não homogêneas nas suas superfícies.

1.1 Motivação

A ampla gama de aplicações de modelos de animação baseados em princípios físicos motivaram a formulação de pesquisa deste trabalho, porque eles dão-se nas visualizações científicas, em simulações de fenômenos naturais, na indústria de entretenimento, em filmes, nas aplicações de imersão da realidade virtual, em mecânica automotiva, etc. Nestes campos, os elementos animados adquirem movimentos regidos pelas leis da física e suas interações permitem que os objetos adquiram comportamentos tal como deveria ocorrer no mundo real.

A geração de comportamentos reais dos objetos, em diversas aplicações, requerem de procedimentos de detecção e tratamento de colisões. Esses procedimentos, em alguns casos, devem ser efetuados em detalhe no que diz respeito às características das superfícies em contato, e, em outros, podem ser tratados sem esta consideração. Um exemplo dessas últimas aplicações é relacionado com os veículos simuladores, tais como os simuladores de vôos, que exigem realismo e precisão física. Neste caso, alguns objetos que estão no ambiente virtual não precisam ser tratados em detalhe, e outros que estão diretamente relacionados com os elementos em análise podem ser tratados em detalhe, inclusive considerando-se as rugosidades de suas superfícies.

¹Irregularidades do detalhe que definem as rugosidades da superfície

Há uma tendência crescente de aplicações de imersão em realidade virtual, onde é muito frequente a interação entre os objetos complexos e agentes externos imersos no ambiente [FucBis92]. Neste tipo de aplicação, os testes e resoluções de interferências devem ser feitos em tempo real. As características fundamentais dos objetos devem ser levadas em conta para se obter efeitos de dinâmica semelhantes aos que acontecem no mundo real.

Simular animações próximas à realidade requer que os objetos do cenário de animação tenham características próprias dos objetos reais, tal como a de possuir geometria arbitrária, com perturbações de suas superfícies. Construir modelos de animação com objetos dessa natureza é um desafio que se pretende abordar neste trabalho.

1.2 Abordagens atuais e suas limitações

Uma arquitetura típica para animação e simulações baseadas em leis da física é composta por três grandes módulos: modelagem dos objetos do ambiente de animação, geração dos movimentos dos objetos (dinâmica de movimento) e controles de simulações (dinâmica de restrições).

Os três módulos da arquitetura da animação, formalizadas por Cremer [Cremer89], estão relacionados com os dois grandes grupos descritos anteriormente. Isto é, o grupo de descrição dos objetos é relacionado com a fase de modelagem dos objetos, e o grupo de comportamento é relacionado com os módulos da dinâmica de movimentos e controle de restrições. A integração desses dois grupos, que são necessários para se obter uma animação perceptivelmente real, não é considerada nas diversas abordagens.

Os modelos de simulações e animações baseados em modelos físicos existentes manipulam objetos simples ou complexos compostos por primitivas simples; eles não consideram as características fundamentais da descrição dos objetos, respeito a sua geometria arbitrária e rugosidades variadas de suas superfícies.

Os módulos de comportamentos são efetuados em cada passo de tempo, repetidas vezes, enquanto dura a animação. A fase de geração de movimento, em geral, é baseada na dinâmica de Newton-Euler, que relaciona o movimento dos corpos com as forças e torques aplicados sobre eles, através de equações diferenciais. Essas equações são integradas numericamente em cada passo de tempo para se computar as novas

posições e velocidades dos objetos em movimento. Nesta situação, instantaneamente estáticas, antes da próxima iteração das equações de movimento, devem ser analisadas as restrições de não interpenetração entre os objetos. Esta última fase consiste em duas sub-fases: detecção e tratamento de interferências. A existência de interferências na animação produz a descontinuidade do comportamento dos objetos em animação para evitar as interpenetrações.

No trabalho de Rivera [Rivera96], formula-se uma arquitetura detalhada de sistemas de animação e descreve-se a implementação de um sistema para corpos rígidos poliedrais, tendo proposto modificações nos métodos abordados por Baraff [Baraff92] e Mirtich [Mirtich96] para a detecção e tratamento de interferências. Nesse trabalho, enfatiza-se que os maiores obstáculos residem na detecção e resolução de interferências de corpos, sendo ainda mais difícil para corpos complexos de geometria arbitrária. Para tal caso, é necessário o emprego de estruturas de dados mais sofisticadas do que as usadas nos trabalhos até agora abordados, e o uso de leis físicas apropriadas.

1.2.1 Detecção de interferências

A verificação de interferências entre objetos em movimento requer atenção especial. Teses de doutorados foram dedicadas somente a procurar modelos de detecção de interferências eficientes e robustas, considerando o princípio de que os objetos no mundo real não se interpenetram. A eficiência e robustez dos modelos dependem da geometria e quantidade dos objetos, e do grau de precisão de teste pela interferência (aproximado ou exato). Em geral, o grau de precisão de cada método depende da aplicação.

As diversas abordagens de detecções de interferências podem ser classificadas em dois grupos: abordagens baseadas em estruturas hierárquicas e abordagens baseadas em cálculos pela interseção. As abordagens baseadas em estruturas precisam de espaços adicionais para a criação, em pré-processamento, de estruturas tipo árvores [Hubbard95, GoLiMa96] para se otimizar o tempo na detecção de interferências, enquanto as abordagens baseadas em cálculos pela interseção usam técnicas de verificação de interseção de objetos a partir das características geométricas do objeto [Hahn88, MoorWhil88, Baraff92, Kamat93, Lin94], não precisa espaços adicionais para gerenciar a detecção de interferências, mas em alguns casos, em particular quando

os objetos são complexos, as comparações geométricas ficam mais demoradas.

O algoritmo incremental de Lin [Lin94] permite a verificação de interferências a partir das características geométricas de objetos de faces poliedrais convexos. Os corpos complexos são tratados como agregados de poliedros convexos ou aproximados por tais agregados. O algoritmo usa as regiões de Voronoi associadas aos vértices, arestas e faces de cada poliedro para a detecção de proximidade. De um modo geral, o algoritmo é eficiente para corpos de composição poliédrica, mas não para corpos limitados por superfícies curvas e geometria arbitrária.

Hubbard [Hubbard95] aborda o problema de interferência, para aplicações interativas, por teste de interseção de esferas hierárquicas envolvendo segmentos do objeto. Para isto, ele contrói uma estrutura denominada árvore esfera (*sphere-tree*) para cada objeto. Cada nó pai cobre o pedaço da superfície coberto pela união dos nós filhos. A árvore é construída em pré-processamento em forma *bottom-up* a partir das pequenas esferas distribuídas cobrindo a superfície do objeto aproximado a poliédrico. Objetos complexos, de geometria arbitrárias, não são considerados neste enfoque.

Outra abordagem baseada em envelopes hierárquicos é a estrutura OBBtree (árvore de caixas envolventes orientadas), aperfeiçoada por Gottschalk et al. [GoLiMa96]. Para criar as caixas envolventes é necessário refinar as faces poliédricas dos objetos em triângulos ou quadriláteros. A detecção de interferências é feita verificando-se a interseção de cada par de caixas, ao percorrer-se a árvore. Objetos complexos de constituição poliédrica são tratados por este algoritmo. No entanto, objetos de características complexas com perturbações irregulares ou formados por superfícies curvas não são considerados por esta abordagem.

As técnicas de detecção de interferências espaciais descritas acima são orientadas a objetos poliédricos, ou aproximações por objetos poliédricos das superfícies paramétricas, que envolvem detecção de interferências espaciais exatas [Hahn88, Baraff92, Lin94, GoLiMa96], ou interferências espaciais aproximadas [Hubbard95]. Todos esses métodos analisam as interferências sem considerar a importância das características particulares da superfície dos objetos.

É possível analisar custos e benefícios de cada técnica existente para se adaptar a interferência de objetos de geometria arbitrária com perturbações de modo a se obter uma nova técnica mais eficiente do que outras. Técnicas que não usam estru-

turas hierárquicas não requerem espaços adicionais para o armazenamento de outras estruturas que não sejam dos objetos, mas o tempo de cálculo pelas interseções é maior, sendo neste sentido ineficiente para manipulação de objetos complexos e de grande escala. Já as técnicas que constroem estruturas hierárquicas envolventes são eficientes na manipulação de objetos em grande escala e podem ser estendidas para objetos complexos. No entanto, o uso de espaço para armazenamento das estruturas dificulta o desempenho quando usados em máquinas relativamente pequenas.

Técnicas eficientes de detecção de interferências podem ser obtidas combinando as duas abordagens: construir as estruturas hierárquicas até um nível razoável de cobertura dos objetos e, depois, para os cálculos de pedaços de objetos já reduzidos pelos envelopes para geometrias mais simples e em menor número, usar as técnicas não hierárquicas. Esta abordagem híbrida, usada neste trabalho, equilibra as vantagens e desvantagens de ambas as abordagens, permitindo uma maior flexibilidade para objetos de geometria complexa.

1.2.2 Resolução de interferências

Existem várias abordagens para a geração das respostas a contatos entre os objetos. Baraff [Baraff92] enfatiza a geração das respostas aos contatos (dinâmicos e estáticos) analiticamente, mediante impulsos e forças de contatos, paradigma amplamente estudado por Lötstedt [Lotstedt81]. Os contatos estáticos são resolvidos pelo método de complementaridade, por tanto seus cálculos são pesados embora fornecer resultados exatos. Mirtich [Mirtich96] enfoca a solução de contato através de micro-colisões, paradigma proposto inicialmente por Hahn [Hahn88]. Assim, todo contato passaria a ser um tipo de colisão, e portanto, resolvido através de impulsos. Este método é mais rápido de se computar através de equações simples, mas não resolve todos os casos de contatos, por exemplo contato estático em plano inclinado. O método de penalidade consegue resolver o problema com um grau adequado de realismo, mas a convergência é lenta [MoorWhil88]. Outro enfoque, formulado por Milenkovic [Milenko96], utilizando dinâmica baseada em posições, resolve o problema rapidamente, mas está longe do fenômeno real.

As abordagens de tratamento de interações de objetos descritas acima são orientadas a objetos de superfícies lisas e homogêneas, envolvendo cálculos relativamente

laboriosos que requerem técnicas sofisticadas de resolução de sistemas de equações. Superfícies com perturbações ou com irregularidades (rugosidades não homogêneas) não são consideradas.

A abordagem baseada em impulsos é mais promissora para a animação e simulação baseadas em leis físicas com objetos complexos, considerando suas limitações, já que um contato entre dois objetos pode envolver vários pontos de contato, para os quais, o uso de outras técnicas poderiam ser demoradas. Esta abordagem também envolve técnicas de resolução de sistemas de equações de segundo ordem, e por isto, será necessário simplificar os cálculos para evitar resolução de grandes sistemas de equações.

1.3 Objeto de geometria complexa

Um corpo complexo, considerado para o propósito deste trabalho, é um objeto de geometria arbitrária com perturbações não-homogêneas sobre sua superfície. Considera-se que a superfície do objeto é formada por curvas paramétricas B-splines cúbicas periódicas. Sendo assim, os pedaços da curva vão limitar o interior do objeto para determinar seus elementos dinâmicos.

A geometria da superfície do objeto, com granulações de forma irregular, é determinada por um grande número de pontos de controle, distribuídos aleatoriamente na superfície deste. Este enfoque é usado na definição de curvas e superfícies variacionais de topologia complexa [WelWit94, Welch95]. No entanto, essa visão é limitada, pois um pedaço de superfície irregular, por mais refinado que pareça, sempre conterà rugosidades, e então, será necessário um método de aproximação para expressar as rugosidades de um nível refinado de superfície.

Pode obter-se um objeto de topologia complexa a partir de uma configuração inicial reposicionando-se, manual ou automaticamente, seus pontos de controle. Para esse processo podem ser usadas as técnicas de manipulação direta local [BarBea89, BarFow93], ou de manipulação direta global [RivCarVe99a]. Em princípio, criar um objeto complexo com técnicas de manipulação direta cai no campo de modelagem de objetos de *forma livre* (Free-Form). Outras técnicas de *deformação forma livre* - *FFD* (Free-Form Deformation) são descritas em [MacJoy96, SedPar86, CoqJan91,

HuHuKa92], com diferentes enfoques, em particular para superfícies de topologias arbitrárias modeladas em B-splines.

Existem, ainda, outras técnicas para modelar objeto de geometria arbitrária baseadas em restrições, como a técnica de cálculo variacional [WelWit94, Welch95, Wesseli96], ou a de modelos físicos [GuLiTa+97] que considera energias e forças aplicadas a um corpo para deformar a superfície.

Os objetos de geometria arbitrária (ou objetos complexos), a serem manipulados neste trabalho, não incluem objetos com buracos ou composição de outros objetos complexos. Além disso, propõe-se desenvolver o projeto somente em 2D. Por isto, vai ser nomeado por *contorno* ou *fronteira* à superfície dos objetos em 2D.

1.4 Proposta de pesquisa

O objetivo do trabalho é estabelecer um esquema para se obter animações visualmente reais considerando as características de detalhe dos objetos complexos. Para isto se desenvolvem técnicas de análise de interferência entre objetos complexos em animação baseada em leis da física. As interferências serão tratadas com uma dinâmica quase exata, de forma a se obter animações realistas em tempo real. Diz-se “quase exata” porque a dinâmica de contatos usada neste trabalho é baseada em micro-colisões e conservação de momentos. No entanto, se fosse baseada em restrições, seria dinâmica exata [Mirtich96]. Para atingir o objetivo, propõem-se vários módulos de pesquisa, onde alguns deles dependem dos outros para se conseguir uma animação dos objetos com as características anteriormente definidas.

Um primeiro módulo da pesquisa é relacionado com a definição dos objetos complexos que permitam definir perturbações nos seus contornos. Para isto, explora-se o *framework* de representação em multiresolução baseado em transformadas wavelets dos objetos definidos no espaço mediante pontos de controle. Com este tipo de transformação, os objetos poderão ser expressos em diferentes níveis de resolução, onde os coeficientes em detalhe poderão permitir a definição das características de detalhe de seus contornos.

Quando os objetos adquirem movimentos regidos pelas leis da física, é necessário uma verificação de interferências. Assim, a detecção de interferências entre objetos

com as características definidas anteriormente requer que o objeto esteja envolvido por estruturas hierarquicamente representadas, compostas por suas partes. A forma de se criar esses envelopes e de representá-los em forma de árvore é outro módulo da pesquisa. Segundo a técnica de geração de envelopes hierárquicos, explora-se a representação em multiresolução do objeto para se criar e se agrupar os envelopes envolventes, conservando-se as características de cada segmento do objeto.

Outra parte que é enfatizada neste trabalho, constituindo um terceiro módulo, é relativa à representação das perturbações mais refinadas das fronteiras dos objetos. Sendo que, devido à impossibilidade de se representar geometricamente todos os detalhes de rugosidades variadas, são estimadas as perturbações a partir de uma biblioteca de variações, e na representação em multiresolução do objeto são explorados estimadores estatísticos de variação de perturbações a partir dos coeficientes de wavelets.

Finalmente, as estruturas hierárquicas de envolventes e os elementos que definem as perturbações nas superfícies dos objetos permitirão a determinação dos pontos de contatos. Para cada um desses pontos será possível calcular a fricção correspondente, para o tratamento de colisões respectivas, com a finalidade de produzir os efeitos do realismo da animação.

A representação dos objetos em multiresolução pode, inclusive, permitir a definição do objeto como sendo refinado em qualquer nível de resolução, sendo que as versões superiores a essa versão podem ser consideradas como parte das perturbações da superfície considerada. Com esta abordagem, podem ser tratadas em detalhe, em alguns casos, resoluções mais refinadas, enquanto em outros casos podem ser tratados com menos detalhe, considerando-se objetos em versões menos refinadas.

1.5 Contribuições

No trabalho que se formula, pretende-se contribuir em vários aspectos no campo da animação baseada em leis da física. As contribuições incluem:

- Novo método para animar objetos de geometria arbitrária, com critérios de dinâmica quase exata e eficiência para animação em tempo real.

- Integração da geometria e perturbações não homogêneas do objeto com a dinâmica.
- Análise de dinâmica de contato entre objetos expressas em diferentes versões.
- Formulação da estrutura OBBtree em multiresolução (MOBBtree) para a detecção de colisões em objetos de geometria arbitrária com perturbações.
- O uso de técnicas de multiresolução wavelets, para a criação de envelopes hierarquicamente estruturados envolvendo partes do objeto.
- Uso de tolerâncias estatísticas, relacionadas com os coeficientes de detalhe da superfície do objeto, para o cálculo de atrito de contato.
- O uso de manipulação direta em multiresolução para a modelagem e deformação da superfície dos objetos complexos.
- Simulação de comportamento real dos objetos em contato, dependendo diretamente da estrutura irregular da fronteira de cada corpo.

1.6 Organização

Para atingir o objetivo, o trabalho é estruturado em oito capítulos. O Capítulo 2 é dedicado à definição dos objetos, sua representação em multiresolução baseada nas transformadas wavelets, e sua manipulação para criar objetos de geometria arbitrária.

O Capítulo 3 é dedicado à criação de envelopes envolventes dos objetos e partes deles, e suas representações em uma estrutura de árvore binária. Neste capítulo são usados segmentos a fronteira do objeto em multiresolução para a definição dos envelopes e a criação da árvore.

No Capítulo 4 são abordadas, em detalhe, as inclusões de perturbações de segmentos de cada nível de resolução nos respectivos envelopes, explicados no capítulo anterior. Neste capítulo são formalizados os aspectos teóricos de perturbações de superfícies em multiresolução.

No capítulo 5 é definida a geração de rugosidades por interpolação de tolerâncias associadas a cada segmento de superfície refinada, com as quais são estimados os coeficientes de fricção correspondente a cada ponto de colisão.

No Capítulo 6 é tratada a geração de movimento dos objetos. Neste capítulo, também, são tratados as interferências espaciais grosseiras, verificação pela interseção dos envelopes da estrutura hierárquica, e finalmente a comparação analítica de segmentos para se obter os respectivos pontos de contato.

No Capítulo 7 aborda-se a dinâmica de colisões baseada em impulsos. Formula-se uma maneira simples de calcular os impulsos correspondentes a cada ponto de colisão tolerado. Também se apresentam alguns resultados experimentais de exemplos de testes validando o modelo formulado neste trabalho.

Por último, no Capítulo 8, são feitas as respectivas conclusões gerais e particulares relativas aos módulos. Os trabalhos futuros também são detalhados neste último capítulo.

Capítulo 2

Objetos de geometria complexa em multiresolução

Entende-se por objetos de geometria complexa como sendo aqueles que não possuem descrição geométrica através de formas simples. Estes corpos podem ser modelados de várias maneiras: por aproximação de faces poligonais, por operações lógicas de primitivas geométricas, etc. Neste capítulo, para o objetivo deste trabalho, um objeto complexo bidimensional é modelado por curvas fechadas B-splines cúbicas. Desta forma, o objeto é limitado por uma seqüência de pedaços de curvas cúbicas. A curva que define o objeto será chamado, daqui para frente, de contorno do objeto, visando a natural extensão para objetos tridimensionais.

A geometria do contorno do objeto pode ser irregular, com granulações determinada por um grande número de pontos de controle, distribuídos aleatoriamente. Sua representação em multiresolução facilita analisar a irregularidade de seu contorno através do detalhe de seu contorno. Para isto, são apresentados os aspectos teóricos de transformadas wavelet, assim como seu uso em multiresolução de curvas.

Um objeto definido pode-se tornar ainda mais complexo manipulando-se o seu contorno (deformando-o). Com este objetivo, propõe-se a manipulação direta de curvas em multiresolução, de forma a conseguir uma deformação real, quase física, do objeto.

Neste trabalho, a definição dos objetos de geometria complexa não considera objetos com buracos ou obtidos através da composição de outros objetos complexos.

2.1 Funções-base B-splines cúbicas periódicas contínuas

As funções-base B-splines apresentadas na literatura, como em [RogAda90, PieTil95, Farin90], podem ser, dependendo da distribuição de seu domínio, contínuas ou não contínuas, periódicas ou não periódicas. Entre elas, as funções bases B-splines periódicas e contínuas facilitam a geração de contornos fechados.

A função-base B-spline, $N_i^k(t)$, de ordem k (grau $k - 1$), é definida recursivamente, através de:

$$N_i^1(t) = \begin{cases} 1 & \text{se } x_i \leq t < x_{i+1} \\ 0 & \text{outros} \end{cases}$$

e

$$N_i^k(t) = \frac{(t - x_i) \cdot N_i^{k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t) \cdot N_{i+1}^{k-1}(t)}{x_{i+k} - x_{i+1}}. \quad (2.1)$$

Os valores x_i , conhecidos como *nós*, são elementos do vetor de nós X , satisfazendo a condição $x_i \leq x_{i+1}$ ($[x_i, x_{i+1})$ é um intervalo de domínio). Quando se constrói uma curva com n pontos de controle, usando estas funções, o número de nós depende de n e k . Assim, o vetor de nós de um intervalo aberto para n pontos de controle tem $n + k$ pontos ($X = [x_0, x_1, x_2, \dots, x_{n+k-1}]$). Em geral, os valores escolhidos para os nós são os números inteiros $0, 1, 2, \dots, n + k + 1$.

As funções-base B-spline uniformes e periódicas não são afetadas pelos nós extremos ou limites. Cada função-base deste tipo é a mesma, só deslocada em uma posição (um nó) sobre o intervalo X :

$$N_{i-1}^k(t - 1) = N_i^k(t) = N_{i+1}^k(t + 1).$$

A Figura 2.1 mostra as funções-base B-spline cúbicas uniformes e periódicas. Em cada intervalo de domínio, a função base é um segmento de uma curva cúbica. Assim, para cada função-base, $N_i^4(t)$, tem-se 4 intervalos de domínio ($[x_i, x_{i+4})$).

As funções-base interpoladas linearmente com os pontos de controle $c_i = (x_i, y_i)$ definem a curva B-splines $f(t)$ de grau $k - 1$, neste caso 3. Desta forma, cada base

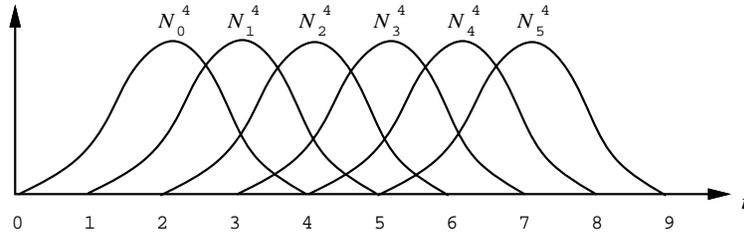


Figura 2.1: Funções-base B-splines cúbica periódicas.

B-spline é associada com um ponto de controle c_i . Tem-se então:

$$f(t) = \sum_{i=0}^n c_i \cdot N_i^k(t). \quad (2.2)$$

Assim, cada ponto afeta uma porção da curva definida pelo domínio $[x_i, x_{i+k})$.

2.1.1 Curvas fechadas

Para $t \in [x_i, x_{i+1})$, aplicando (2.2) obtém-se um segmento de $f(t)$ envolvendo somente k bases que tem como domínio $[x_i, x_{i+1})$. O problema de domínios próximos aos extremos são resolvidos considerando os nós de modo circular, de forma que os nós seguintes ao último nó, x_{n-1} , são os primeiros nós, e os anteriores aos primeiros nós são os últimos nós. Com isto, para um domínio definido por $X = [x_0 \ x_1 \ \dots \ x_{n-1}]$, o domínio para a curva fechada é concebido como sendo $X^f = [\dots x_{n-2} \ x_{n-1} \ x_0 \ x_1 \ \dots \ x_{n-1} \ x_0 \ x_1 \ \dots]$.

A Figura 2.2 mostra uma curva fechada com oito pontos de controle. Observe-se que a curva fechada é obtida repetindo-se os primeiros 3 pontos de controle nas últimas posições.

Na geração de uma curva fechada f com n pontos de controle, cada ponto da curva $f(t)$ só envolve k funções bases significativas chamadas *funções base relativas ao ponto*. Interpolar apenas k elementos para calcular um ponto sobre a curva é computacionalmente atraente para o propósito deste trabalho, já que se espera manipulação de um grande número de curvas fechadas, cada curva tendo um grande número de pontos de controle. Com as mesmas k funções base são computados todos os pontos do segmento de curva $f_j(t)$, para $t \in [x_{j+k}, x_{j+k+1})$ do domínio. Com esta visão, a

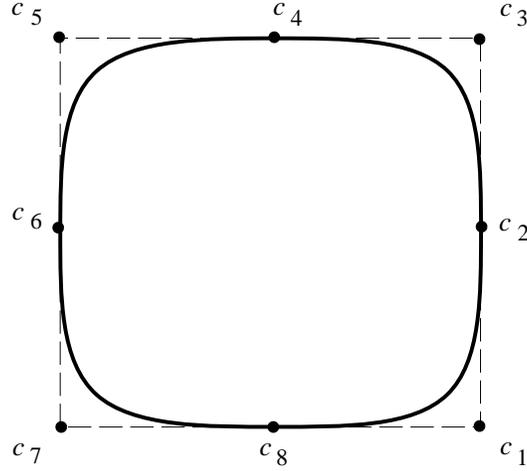


Figura 2.2: Uma curva B-spline cúbica com oito pontos de controle.

expressão (2.2) pode-se escrever como

$$f_j(t) = \sum_{i=0}^{k-1} c_{j+i-1} N_{i,j}^k(t), \quad \text{com } t \in [x_{j+k}, x_{j+k+1}) \quad \text{para } j = 1, \dots, n. \quad (2.3)$$

Se todos os intervalos do domínio $[x_{j+k}, x_{j+k+1})$ são de tamanho constante, as bases podem ser representadas matricialmente ([RogAda90]). Para este propósito, o domínio de parâmetros deve ser reparametrizado para $[0, 1)$. As k funções base relativas aos pontos do segmento da curva $f_j(t)$, também reparametrizadas, são denotadas pela matriz $\tilde{N}^k(t)$, e os k pontos de controle relativos a estas funções estão representados em um vetor C_j^k . Com estes critérios, a expressão (2.3) é equivalente a

$$f_j(t) = \tilde{N}^k(t) \cdot C_j^k, \quad (2.4)$$

onde $\tilde{N}^k(t) = \frac{1}{(k-1)!} [t^{k-1} \ t^{k-2} \ \dots \ 1] [\tilde{n}_{i,r}^k]$ com $0 \leq t < 1$. Os elementos $\tilde{n}_{i,r}^k$ são inteiros, e são obtidos pela fórmula de Cohen e Riesenfeld [RogAda90]. Especificamente para $k = 4$, a matriz $[\tilde{n}_{i,r}^k]$ gerada é

$$[\tilde{n}_{i,r}^4] = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix}, \quad (2.5)$$

onde cada coluna r da matriz é formada pelos coeficientes da função base reparametrizada

trizada $\tilde{N}_r^4(t)$. A matriz coluna C_j^4 , de 4 pontos de controle, tem a forma

$$C_j^4 = [c_{j-1} \ c_j \ c_{j+1} \ c_{j+2}]^T.$$

Observa-se que a expressão $\tilde{N}^k(t)$ não depende de j , porque é constante para todos os intervalos de domínio. A Figura 2.3 mostra o comportamento das quatro funções bases em um intervalo de domínio reparametrizado para $[0, 1)$, os quais vão ser usados para cômputo de todos os segmentos f_j da curva.

O segmento de curva $f_j(t)$ é gerado por C_j^4 ; por isto, considera-se pontos de controle c_j e c_{j+1} como sendo relativos ao segmento. Desta forma, para n pontos de controle existem n segmentos de curva.

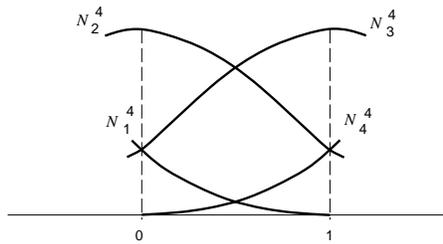


Figura 2.3: Segmentos de funções bases B-spline cúbica reparametrizadas para intervalo $[0, 1)$.

Um objeto complexo, bidimensional, de forma arbitrária, será definido por uma curva fechada (formulação (2.4)), que representa o contorno do objeto. A Figura 2.4 é um exemplo de um objeto com 32 pontos de ontrole.

2.1.2 Derivadas das funções B-Splines periódicas

Com a derivada da curva obtém-se a reta tangente passando por um ponto da curva, e com a segunda derivada verifica-se a variação da tangente. Estes parâmetros do comportamento da curva serão usados em diversas fases do presente trabalho.

A forma matricial permite calcular, de forma simples, as derivadas em um ponto qualquer da curva. Como uma curva B-spline de grau $k - 1$ é derivável $k - 2$ vezes, a curva B-spline cúbica, em particular, é contínua até a segunda derivada.

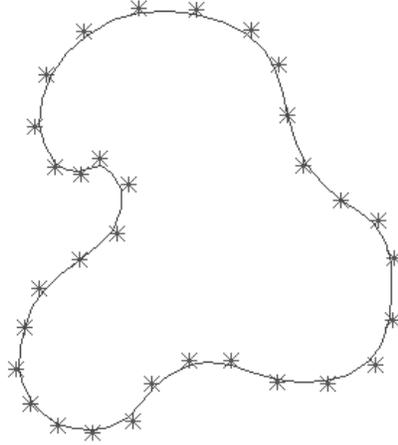


Figura 2.4: Um objeto complexo limitado por uma curva fechada B-spline cúbica com 32 pontos de controle.

A primeira derivada do segmento $f_j(t)$ é dada por

$$f'_j(t) = \tilde{N}^{k-1}(t) \cdot C_j^{k-1}. \quad (2.6)$$

Para $k = 4$, as matrizes são:

$$\tilde{N}^3(t) = \frac{1}{2} [t^2 \ t \ 1] \begin{pmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{pmatrix} \quad \text{e} \quad C_j^3 = \begin{pmatrix} c_j - c_{j-1} \\ c_{j+1} - c_j \\ c_{j+2} - c_{j+1} \end{pmatrix}.$$

A segunda derivada é dada por

$$f''_j(t) = \tilde{N}^{k-2}(t) C_j^{k-2}. \quad (2.7)$$

Para $k = 4$, as matrizes são:

$$\tilde{N}^2(t) = [t \ 1] \begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{e} \quad C_j^2 = \begin{pmatrix} c_{j+1} - 2c_j + c_{j-1} \\ c_{j+2} - 2c_{j+1} + c_j \end{pmatrix}.$$

Para maiores detalhes na dedução das derivadas, ver [RogAda90, PieTil95].

2.2 Representação em multiresolução

Seja $f(t)$ a função que define o contorno de um objeto complexo, com 2^n pontos de controle escritos em uma matriz coluna $C^n = [c_0^n \ c_1^n \ c_2^n \ \dots \ c_{2^n-1}^n]^T$, onde cada elemento

c_s^n , no plano, é um ponto de coordenadas x e y ($c_s^n = (x_s^n, y_s^n) \in R^2$). Para efeitos de sua representação em multiresolução, esta função, com estas características, e definida como descrita na Seção 2.1.1, será denotada por $f^n(t)$ ou simplesmente por $f(t)$, onde n representa o nível de resolução. Neste caso f^n é a função limite representa o contorno refinado do objeto, enquanto as outras $f^j(t)$, com $j < n$, serão representações grosseiras do objeto em versão j . Em alguns casos, em particular quando tenha que se referir a vários objetos em multiresolução, será denotado por f^{-j} , para $j = 1, 2, \dots$, como resoluções grosseiras do nível j . Para se denotar contornos de resoluções mais refinados do que definidos geometricamente, como descritas na Seção 2.4 e no Capítulo 6, se usará f^{+1}, f^{+2}, \dots para denotar resoluções refinadas de primeiro nível, segundo nível, etc.

Um objeto de uma resolução j pode ser obtido a partir dos pontos de controle dessa versão, e sua diferença do f^{j+1} constitui a curva de detalhe $o^j = f^{j+1} - f^j$. Os pontos de controle que definem f^j e o^j constituem os coeficientes de escala e os coeficientes de detalhe, respectivamente, do nível j . Desta forma se usará a teoria de wavelets para a representação de um objeto em multiresolução. Na seção a seguir formulam-se, sem entrar em maiores detalhes da teoria de wavelets, os resultados relevantes para se obter objetos em várias resoluções.

2.2.1 Decomposição e composição do objeto em várias resoluções

Para se obter uma função f^{j-1} , é necessário determinar a matriz coluna C^{j-1} de pontos de controle do objeto a partir de C^j . O filtro para este processo é denominado filtro de análise, dado por uma matriz de constantes A^j de dimensão $2^{j-1} \times 2^j$, que é chamada também “matriz de análise de coeficientes”, de forma que

$$C^{j-1} = A^j C^j \quad (2.8)$$

Os coeficientes de detalhe, desta resolução, são capturados em uma outra matriz coluna $D^{j-1} = [d_0^{j-1} \ d_1^{j-1} \ d_2^{j-1} \ \dots \ d_{2^{j-1}-1}^{j-1}]^T$, usando-se a matriz B^j de dimensão $2^{j-1} \times 2^j$ chamada a *matriz de análise de detalhe*, como

$$D^{j-1} = B^j C^j. \quad (2.9)$$

O número de elementos de D^{j-1} é o mesmo de C^{j-1} , ou seja 2^{j-1} . As matrizes A^j e B^j são chamadas “matrizes de filtro de decomposição”.

O processo de decomposição de C^j em C^{j-1} e D^{j-1} , para $\min \leq j \leq n$ (sendo \min o nível mínimo requerido para manter a forma de um objeto), é aplicado recursivamente sobre C^{j-1} . Em particular deve-se impor a condição $\min \geq 4$ para manter a curva fechada.

Para a composição do objeto a partir dos coeficientes relativos às representações de resolução menor que n , cujo processo é conhecido como transformada inversa, se usarão as *matrizes de filtro de composição ou síntese* P^j e Q^j , cujas dimensões são $2^j \times 2^{j-1}$, sendo a sua formulação

$$C^j = P^j C^{j-1} + Q^j D^{j-1}. \quad (2.10)$$

A Figura 2.5 mostra a decomposição dos coeficientes C^j em C^{j-1} e D^{j-1} e a composição de C^{j-1} e D^{j-1} em C^j .

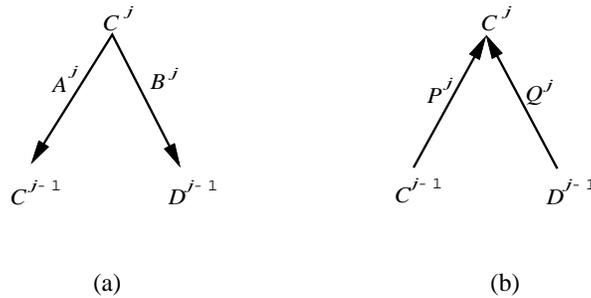


Figura 2.5: Estrutura de transformação direta e inversa:(a) decomposição; (b) composição.

O número de elementos de C^n deve ser uma potência de 2, já que o processo de decomposição sempre vai gerar metade dos pontos em cada recursão.

As matrizes de filtro A^j , B^j , P^j e Q^j são calculadas associando a função que define $f^j(t)$ com os coeficientes de escala C^j , para $0 \leq t \leq 1$, tal como definido pela expressão (2.4).

$$f^j(t) = \Phi^j(t) C^j. \quad (2.11)$$

O vetor $\Phi^j(t) = [\phi_1^j(t), \dots, \phi_{2^j}^j(t)]$ é chamado *função base de escala*. Neste caso, é um vetor das funções B-splines cúbicas definidas na Seção 2.1, e é refinável pela matriz

de filtro P^j ,

$$\Phi^{j-1}(t) = \Phi^j(t)P^j. \quad (2.12)$$

As bases Φ^j geram o espaço V^j , as funções refinadas Φ^{j-1} geram o espaço V^{j-1} tal que $V^{j-1} \subset V^j$, e assim recursivamente. Do mesmo modo, os espaços complementos W^{j-1} , tal que $V^{j-1} \oplus W^{j-1} = V^j$, são gerados por *funções base wavelets* $\Psi^{j-1}(t)$,

$$\Psi^{j-1}(t) = \Phi^j(t)Q^j. \quad (2.13)$$

Com estas formulações obtém-se a expressão que relaciona os quatro filtros A^j , B^j , P^j e Q^j , que finalmente se expressam como

$$\begin{bmatrix} A^j \\ B^j \end{bmatrix} = [P^j | Q^j]^{-1}. \quad (2.14)$$

Desta forma, dados os valores dos filtros de síntese, é imediato calcular os valores dos filtros de análise. Os filtros P^j e Q^j são calculados definindo o produto interno entre as bases, de forma que V^{j-1} e W^{j-1} sejam ortogonais [StDeSa96].

Os espaços $\{V^j\}$, tal que $\dots \subset V^{j-1} \subset V^j \subset \dots$, definem uma representação em multiresolução de f , já que a sua projeção sobre cada V^j é f^j (função de nível j). A definição formal de representação em multiresolução é dada em [GomVel98].

Um função f pode ser expressa em termos de seus coeficientes wavelets e sua projeção no nível mínimo, como

$$f = f^{min} + \sum_{i=min}^n \Phi^i D^i. \quad (2.15)$$

O uso das funções base B-splines periódicas na definição dos objetos fechados permite que todas as colunas das matrizes de filtros tenham elementos comuns deslocados em alguma posição. Isto vai evitar a manipulação de grandes matrizes, tanto na análise quanto na síntese. Na seguinte seção se estabelece a relação dos elementos das matrizes de filtros e os coeficientes.

2.2.2 Relação de coeficientes e filtros

Uma função de escala $\phi_s^j(t)$ é definida pelas operações de translação e dilatação de uma “função mãe” $\phi(t)$, como explicado em diversos trabalhos, tais como [StDeSa96, SweSch95, GomVel98], como

$$\phi_s^j(t) = \phi(2^j t - s). \quad (2.16)$$

Daqui a relação de um nível para o outro da função mãe é expressa usando um coeficiente de filtro p_m , como em (2.12):

$$\phi(t) = \sum_m p_m \phi(2t - m). \quad (2.17)$$

O processo de refinamento das funções base obtém-se substituindo (2.17) em (2.16),

$$\begin{aligned} \phi_s^j(t) &= \sum_m p_m \phi(2^j t - s - m) \\ &= \sum_m p_m \phi_{2^j s - m}^{j+1}(t). \end{aligned} \quad (2.18)$$

A qual é equivalente, fazendo uma mudança nos índices, a

$$\phi_s^j(t) = \sum_m p_{m-2^j s} \phi_m^{j+1}(t). \quad (2.19)$$

Do mesmo modo, refinando-se a função $\psi_s^j(t)$ e operando, chega-se à expressão

$$\psi_s^j(t) = \sum_m q_{m-2^j s} \phi_m^{j+1}(t). \quad (2.20)$$

Como a síntese constrói a projeção da função f sobre um espaço, por exemplo, V^{j+1} , este seria da forma

$$\begin{aligned} f^{j+1} &= f^j + o^j \\ &= \sum_s c_s^j \phi_s^j + \sum_s d_s^j \psi_s^j \\ &= \sum_s c_s^j \left(\sum_m p_{m-2^j s} \phi_m^{j+1} \right) + \sum_s d_s^j \left(\sum_m q_{m-2^j s} \phi_m^{j+1} \right) \\ &= \sum_m \left(\sum_s p_{m-2^j s} c_s^j + \sum_s q_{m-2^j s} d_s^j \right) \phi_m^{j+1} \\ &= \sum_m c_m^{j+1} \phi_m^{j+1} \end{aligned}$$

Em conseqüência, obtém-se a relação dos coeficientes de composição, como

$$c_m^{j+1} = \sum_s p_{m-2^j s} c_s^j + \sum_s q_{m-2^j s} d_s^j. \quad (2.21)$$

Critério similar será usado para se obter a relação dos filtros e coeficientes de decomposição (análise), chegando-se às expressões:

$$c_m^j = \sum_s a_{s-2m} c_m^{j+1} \quad (2.22)$$

e

$$d_m^j = \sum_s b_{s-2m} c_m^{j+1}. \quad (2.23)$$

Observe-se que, nos cálculos dos coeficientes c_m^j e d_m^j , são capturados elementos pares de C^{j+1} , pela existência de $2m$ nas expressões. Este é uma operação de decimação. Quer dizer de que em cada recursão de decomposição são gerados a metade dos coeficientes de C^{j+1} . Os elementos a, b, p e q são elementos de uma coluna das matrizes A, B, P e Q respectivamente, e não dependem do nível j .

Os valores dos filtros são calculados definindo o produto interno, de tal forma que as bases ϕ^j e ψ^j sejam ortogonais, cumprindo a condição de espaços de multiresolução $V^j \oplus W^j = V^{j+1}$ (definidos na seção anterior). Estes valores podem ser calculados pelo método de lifting [SweSch95], ou pelo método de domínio de frequências [GomVel98]. Estes algoritmos não são abordados neste trabalho porque não se pretende construir análise multiresolução, mas somente usá-la. Os valores dos filtros usados são apresentados no Apêndice A, para transformações wavelets biortogonais, que são tratados brevemente na seção a seguir.

2.2.3 Multiresolução Biortogonal

Wavelets biortogonais constituem uma generalização de wavelets ortogonais. A classe das wavelets biortogonais tem qualidades da classe wavelets ortogonais e é mais flexível que ela. Estas características são fundamentais para expressar um objeto complexo em multiresolução, para a manipulação de seu contorno.

Em multiresolução biortogonal, são usadas as bases biortogonais ao invés das ortogonais, sendo uma para análise e outra para a síntese. Isto é, para o cálculo de c_s^{j-1} e d_s^{j-1} são usadas as bases *primais* ϕ_s^j e ψ_s^j , respectivamente, e para síntese são usadas as bases *duais* $\tilde{\phi}_s^j$ e $\tilde{\psi}_s^j$:

$$f^{j-1} = \sum_m c_s^{j-1} \tilde{\phi}_s^{j-1} \quad e \quad o^{j-1} = \sum_m d_s^{j-1} \tilde{\psi}_s^{j-1}.$$

As funções ϕ_s^j e $\tilde{\phi}_s^j$ são bases dos espaços V^j e \tilde{V}^j , respectivamente, assim como ψ_h^j e $\tilde{\psi}_s^j$ são bases de W^j e \tilde{W}^j . Desta forma, tem-se dois espaços de multiresolução

$$\begin{aligned} \dots &\subset V^{j-1} \subset V^j \subset \dots \\ \dots &\subset \tilde{V}^{j-1} \subset \tilde{V}^j \subset \dots, \end{aligned}$$

com

$$V^{j-1} + W^{j-1} = V^j \quad e \quad \tilde{V}^{j-1} + \tilde{W}^{j-1} = \tilde{V}^j.$$

A ortogonalidade ocorre entre espaços primais e duais

$$V^j \perp \tilde{W}^j,$$

$$\tilde{V}^j \perp W^j$$

e

$$W^j \perp \tilde{W}^i, \quad \text{com } j \neq i.$$

Isto implica que seus produtos internos sejam da forma

$$\langle \tilde{\phi}(t), \psi(t-s) \rangle = 0$$

$$\langle \tilde{\psi}(t), \phi(t-s) \rangle = 0$$

e

$$\langle \tilde{\phi}(t), \phi(t-s) \rangle = \delta_s$$

$$\langle \tilde{\psi}(t), \psi(t-s) \rangle = \delta_s.$$

Com estas características de biortogonalidade das bases, as relações dos filtros com as bases, são dadas de forma análoga às expressões (2.19) e (2.20), como

$$\phi(t) = \sum_s h_s \phi(2t-s) \quad \text{e} \quad \psi(t) = \sum_s g_s \phi(2t-s).$$

Similarmente

$$\tilde{\phi}(t) = \sum_s \tilde{h}_s \tilde{\phi}(2t-s) \quad \text{e} \quad \tilde{\psi}(t) = \sum_s \tilde{g}_s \tilde{\phi}(2t-s).$$

Finalmente, combinando estas expressões com os produtos definidos acima, obtém-se a relação dos coeficientes e filtros, para decomposição

$$c_m^j = \sum_s h_{s-2m} c_s^{j+1}, \tag{2.24}$$

$$d_m^j = \sum_s g_{s-2m} c_s^{j+1} \tag{2.25}$$

e para composição

$$c_m^{j+1} = \sum_s \tilde{h}_{m-2s} c_s^j + \sum_s \tilde{g}_{m-2s} d_s^j. \tag{2.26}$$

Observa-se que, na decomposição, às expressões (2.22) e (2.23) são idênticas a (2.24) e (2.25), respectivamente. A mesma característica se observa no processo de composição entre as expressões (2.21) e (2.26). Neste sentido, para efeitos de manipulação de decomposição e composição dos corpos, os coeficientes de filtro são

considerados como $a = h$, $b = g$, $p = \tilde{h}$ e $q = \tilde{g}$. Os coeficientes de relação podem ser intercambiados; assim, dados os filtros, calculados em termos de multiresolução biortogonal B-splines cúbico, apresentados no Apêndice A, pode se operar sem problema tanto no processo de análise quanto no de síntese, pelo processo de transformada rápida formulada pelas expressões (2.28) e (2.29) na seção a seguir. O método de cálculo dos filtros para este tipo de multiresolução é apresentado em [GomVel98].

Um critério importante considerado nos cálculos dos filtros, em particular para transformadas wavelets biortogonais [Sweldens96, SweSch95], é que os coeficientes de escala C^j representam a média dos coeficientes de escala C^{j+1} . Desta forma, a média dos coeficientes de detalhe D^j tem tendência a zero. Com isto, a representação de f no nível j , ou seja f^j , é representativa, porque a perda em detalhes de um nível para outro é mínima.

2.2.4 Pontos de controle do objeto em multiresolução

Em vista que uma curva fechada B-spline é definida por bases B-splines periódicas e contínuas, cada ponto da curva fechada sempre tem pontos vizinhos em ambos os lados, não existindo, assim, pontos extremos. Nos pontos de controle $C = \{c_1, c_2, \dots, c_n\}$, os pontos c_1 e c_n (supostamente extremos) são vizinhos. Sendo assim, a seqüência de pontos é cíclica já que sempre um ponto de controle tem vizinho esquerdo e direito, tal como ilustrado pela Figura 2.6.

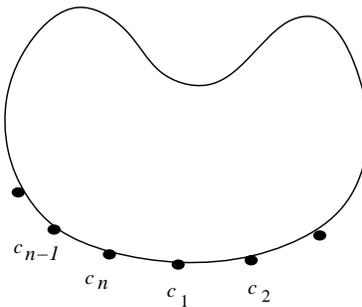


Figura 2.6: Um objeto com n pontos de controle, tratados em ciclo.

Para os efeitos de análise, consideram-se os elementos significativos das matrizes de filtro, em que cada coluna é simétrica, e se repetem em cada coluna com um

computado como

$$c_s^{j-1} = a_0 c_{2s}^j + \sum_{i=1}^r a_i (c_{2s-i}^j + c_{2s+i}^j), \quad (2.27)$$

e um elemento d_s^{j-1} de D^{j-1} é calculado como

$$d_s^{j-1} = b_0 c_{2s-1}^j + \sum_{i=1}^x b_i (c_{2s-i}^j + c_{2s+i}^j). \quad (2.28)$$

Para o processo de composição (reconstrução), de modo análogo à decomposição, pode-se simplificar a operação da expressão (2.10) em um cálculo linear simples, adequando a expressão (2.21) como

$$c_{2s}^j = p_0 c_s^{j-1} + \sum_{i=1}^v p_{2i} (c_{s-i}^{j-1} + c_{s+i}^{j-1}) + \sum_{i=0}^w q_{2i-1} (d_{s-i}^{j-1} + d_{s+i+1}^{j-1})$$

e (2.29)

$$c_{2s+1}^j = \sum_{i=0}^v p_{2i-1} (c_{s-i}^{j-1} + c_{s+i+1}^{j-1}) + q_0 d_{s-1}^{j-1} + \sum_{i=1}^w q_{2i} (d_{s-i+1}^{j-1} + d_{s+i+1}^{j-1}).$$

As operações simplificadas, expressas por (2.27) e (2.28) para decomposição, e (2.29) para a reconstrução, são apresentadas em [GomVel98] em forma de um segmento de código em C como a *transformação rápida de wavelets*. Também se discute o grau de complexidade, que é linear em relação ao número de coeficientes m e o número de elementos de filtro r , sendo assim $O(mr)$. A Figura 2.7 mostra a decomposição em dois níveis de um objeto com 32 pontos de controle.

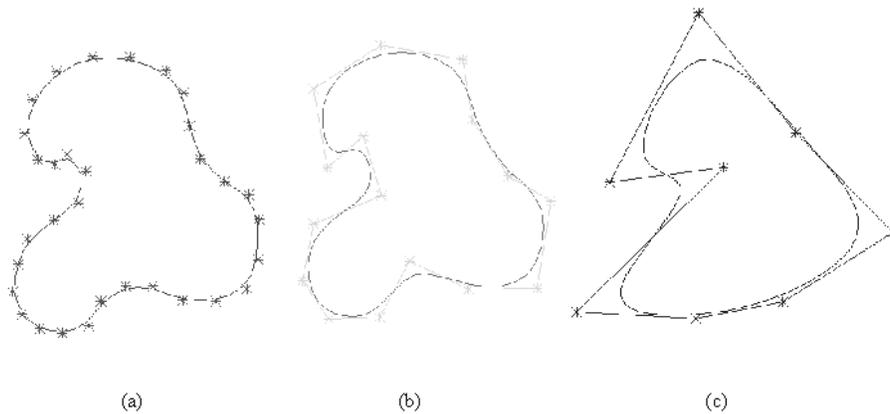


Figura 2.7: *Decomposição de um objeto em dois níveis: (a) objeto original; (b) objeto decomposto no primeiro nível; (c) objeto decomposto no segundo nível.*

2.2.5 Refinamento dos pontos de controle do objeto

Refinar pontos de controle de um objeto consiste em inserir mais pontos de controle para o mesmo objeto, sem alterar a sua forma. Existem abordagens em curvas B-splines para este problema, via inserção de nós seqüencial ou paralela [PieTil95]. Mas usar os métodos de reconstrução em multiresolução é mais fácil, em especial para o propósito deste trabalho.

Um número maior de pontos de controle permitirá a definição de uma característica irregular do contorno do objeto, já que o inverso, minimização por decomposição, permite sua suavização.

Pode-se obter um objeto refinado, sem usar inserção de nós, aproveitando a técnica de composição de objetos em multiresolução, já que (2.18) permite o refinamento das bases. Isto é, dado o contorno definida pela curva $f^n(t)$, supostamente refinada, com 2^n pontos de controle, pode-se refinar mais ainda para $f^{n+1}(t)$ com 2^{n+1} pontos de controle, representando o mesmo contorno, para o qual é necessário considerar D^n nulo. Assim, cada objeto definido no sistema pode ser refinado para o número de pontos desejados.

A Figura 2.8 mostra o refinamento de um objeto de 32 pontos de controle para ter 64 pontos. A morfologia do objeto continua sendo a mesma.

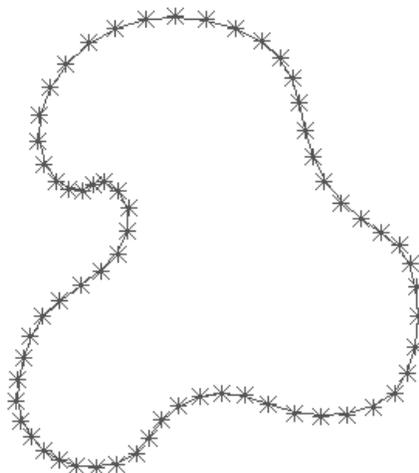


Figura 2.8: Um objeto inicial de 32 pontos de controle refinado para 64 pontos.

2.3 Alteração de forma do objeto

A necessidade de alterar a forma de um objeto é importante em muitas aplicações. Na modelagem pode se obter objetos ainda mais complexos alterando sua forma inicial, mantendo alguma característica deste, tal como número de pontos de controle fixo.

Existem várias técnicas para deformar curvas e superfícies: as técnicas baseadas nas operações sobre os pontos de controle (inserção, eliminação e mudança de posições) [Piegl89, PieTil95, RogFog89, ForBar88], e na variação dos pesos em caso de NURBS [Piegl89, PieTil95]; técnicas de manipulação direta ([BarBea89, BarFow93]), que permitem atuar diretamente sobre o ponto da curva ou de superfície; técnicas empregando cálculo variacional [Welch95, Wesseli96]; técnicas baseadas em modelos físicos [GuLiTa+97, HsHuKa92, LycMor87] que emprega energias na deformação da superfície do corpo.

Em princípio, criar um objeto complexo com técnicas de deformação por manipulação direta cai no campo de modelagem de objetos de *forma livre* (Free-Form), como as técnicas de *deformação forma livre - FFD* (Free-Form Deformation) descritas em [MacJoy96, WelWit94, SedPar86, CoqJan91, HuHuKa92], cada um com diferentes abordagens, em particular para superfícies de geometrias arbitrárias modeladas em B-splines. A técnica de manipulação direta é mais amigável que as técnicas de FFD propriamente ditas.

2.3.1 Manipulação direta do contorno

A mudança de posição do ponto $f_j(t)$ definida por (2.4) acarreta o reposicionamento dos 4 pontos de controle associados a ele (tratados na Seção 2.1.1). Bartels, Beaty e Fowler abordam a técnica de reposicionar cada ponto de controle c_{j+i-1} minimamente, através de Δc_{j+i-1} , de forma que permita levar o ponto $f_j(t)$ para sua nova posição, com a expressão seguinte:

$$\Delta C_j = \Delta f_j(t) N^T (N \cdot N^T)^{-1}, \quad (2.30)$$

onde $\Delta C_j = [\Delta c_{j-1} \Delta c_j \Delta c_{j+1} \Delta c_{j+2}]$, e N é a matriz de bases B-splines composta por quatro bases reparametrizadas, dada por

$$N = \begin{bmatrix} N_1^4(t) \\ N_2^4(t) \\ N_3^4(t) \\ N_4^4(t) \end{bmatrix} = \frac{1}{6} [t^3 \ t^2 \ t \ 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}. \quad (2.31)$$

A Figura 2.9 mostra a deformação de três segmentos do contorno de um corpo complexo definido por 32 pontos de controle.

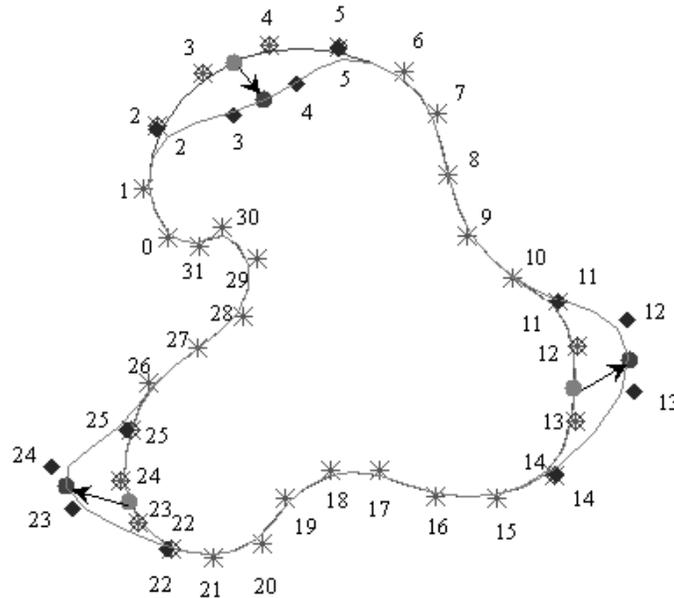


Figura 2.9: Uma curva fechada B-spline cúbica deformada em três pontos de diferentes segmentos da curva.

Um corpo complexo definido por pontos de controle de vizinhança mínima não vai permitir uma deformação substancial do contorno. Sua ação será simplesmente sobre uma pequena porção desta, causando uma deformação local. Ela pode ser pronunciada, acarretando reposicionamentos de uma vizinhança maior que $k = 4$ pontos de controle, o qual é conseguido utilizando a multiresolução de contornos, que são discutidos na próxima seção.

2.3.2 Deformação do objeto em múltiplas resoluções

Aproveita-se a idéia de edição de objetos em multiresolução, comentados brevemente em [FinSal94, KazElb97], para gerar uma deformação da forma do objeto fechado em multiresolução. Esta deformação é obtida pela manipulação direta de um ponto projetado nas resoluções de níveis inferiores da curva. O resultado se percebe na versão refinada após sua reconstrução.

Seja f^n o contorno refinado do corpo, e seja f^j a projeção de f^n no espaço V^j . Uma manipulação direta no ponto $f^j(t)$ em $\Delta f^j(t)$ se propaga através das resoluções de níveis superiores, para produzir uma resolução refinada deformada \tilde{f}^n , como

$$\begin{aligned}\tilde{f}^j(t) &= f^j(t) + \Delta f^j(t), \\ \tilde{f}^{j+1}(t) &= \tilde{f}^j(t) + o^j(t), \\ &\dots \\ \tilde{f}^n(t) &= \tilde{f}^{n-1}(t) + o^{n-1}(t).\end{aligned}$$

A função o^j é a curva de detalhe do nível j .

Neste processo, a alteração é feita nos coeficientes C^j , já que as funções bases são inalteráveis. Assim, no processo de reconstrução, as variações se refletem nos coeficientes respectivos (\tilde{C}^{j+1}).

$$\tilde{C}^j = C^j + \Delta C^j \quad \text{implica} \quad \tilde{C}^{j+1} = P^{j+1}(C^j + \Delta C^j) + Q^{j+1}(D^j).$$

Quando se reposicionam os coeficientes de escala C^j , as magnitudes dos coeficientes de wavelet D^j permanecem fixos, mas eles devem ser reorientados para manter a característica da curva original. Para cada vetor de detalhe d_i^j de D^j é fixada sua orientação respeito a um sistema local de referência definida pela reta tangente (derivada) e sua normal no ponto médio do segmento do contorno respectivo (ver detalhe em [RivCarVe99a]).

Em geral, quando se deforma um objeto, as mudanças que ocorrem nas resoluções inferiores e interiores do objeto devem ser transparentes para o agente modificador. Assim, o agente vai deslocar um ponto $f^n(t)$, do contorno mais refinado, em um vetor $\Delta \mathbf{d}$, possivelmente simulando um material com certa rigidez, mas ele não precisa saber que se deve modificar em alguma porção as camadas interiores, ou outras resoluções, do objeto.

A propagação de deformação em vários níveis superiores de resolução de um contorno depende de uma função distribuidora de segmentos do vetor $\Delta \mathbf{d}$, para serem usados na manipulação direta, automaticamente, nessas resoluções. A função distribuidora $g(s)$, que simula o parâmetro de rigidez do material que define o contorno, tem a forma exponencial negativa para definir segmento de vetor $\Delta \mathbf{d}^n$ para f^n , $\Delta \mathbf{d}^{n-1}$ para f^{n-1} , sucessivamente até $\Delta \mathbf{d}^{n-r}$ para f^{n-r} , de forma que

$$\|\Delta \mathbf{d}^n\| \geq \|\Delta \mathbf{d}^{n-1}\| \geq \dots \|\Delta \mathbf{d}^{n-r}\|.$$

Adotou-se a função $g(s)$ da forma

$$g(s) = \frac{1}{A} e^{-ws}, \quad \text{onde} \quad A = \int_0^1 e^{-ws} ds,$$

no intervalo $0 \leq s < 1$, de modo que este determina a porção de segmento de vetor

$$\Delta \mathbf{d}^i = \left(\int_{\frac{i}{m}}^{\frac{i+1}{m}} g(t) dt \right) \Delta \mathbf{d}.$$

Maiores detalhes deste procedimento podem-se consultar em [RivCarVe99b].

O critério de distribuição de deslocamentos nos diferentes níveis de resolução do objeto induz maior segmento de deformação nas resoluções superiores. Ao se aplicar estes deslocamentos, por manipulação direta do objeto em multiresolução, o ponto $f^n(t)$ é deslocado pelo vetor $\Delta \mathbf{d}$, como mostram os teoremas formulados a seguir.

Teorema 2.1 *O ponto $f^n(t)$ deslocado por um vetor $\Delta \mathbf{d}$ é um outro ponto $p = f^n(t) + \Delta \mathbf{d}$ equivalente ao deslocamento do ponto $f^{n-1}(t_1)$, projeção de $f^n(t)$ em f^{n-1} , deslocado pelo mesmo vetor $\Delta \mathbf{d}$ projetado em f^n .*

Prova: Seja $\Delta \mathbf{d} = \Delta f^n(t)$ o vetor de deslocamento aplicado sobre $f^n(t)$. A função f^n é reconstruída a partir dos coeficientes de baixa resolução como

$$f^n(t) = \Phi^n(t)(P^n C^{n-1} + Q^n D^{n-1}).$$

Se se desloca um ponto de f^{n-1} , projeção de a $f^n(t)$ sobre f^{n-1} , em $\Delta \mathbf{d}$, alguns coeficientes de C^{j-1} são modificados, segundo a fórmula de manipulação direta (2.30), em ΔC^{n-1} . Depois disto, a expressão acima passa a ser

$$\tilde{f}^n(t) = \Phi^n(t) \left(P^n \left(C^{n-1} + \Delta C^{n-1} \right) + Q^n D^{n-1} \right).$$

A expressão (2.30) pode-se reescrever como $\Delta C^{n-1} = \Delta \mathbf{d} \frac{\Phi^{n-1}}{\Phi^{n-1} \cdot \Phi^{n-1}}$. A curva deformada é reescrita como

$$\begin{aligned} \tilde{f}^n(t) &= \Phi^n(t) \left(P^n \left(C^{n-1} + \Delta \mathbf{d} \frac{\Phi^{n-1}}{\Phi^{n-1} \cdot \Phi^{n-1}} \right) + Q^n D^{n-1} \right) \\ &= \Phi^n(t) (P^n C^{n-1} + Q^n D^{n-1}) + \Phi^n(t) P^n \Delta \mathbf{d} \frac{\Phi^{n-1}}{\Phi^{n-1} \cdot \Phi^{n-1}} \\ &= \Phi^n(t) (P^n C^{n-1} + Q^n D^{n-1}) + \Phi^{n-1} \Delta \mathbf{d} \frac{\Phi^{n-1}}{\Phi^{n-1} \cdot \Phi^{n-1}} \\ &= f^n(t) + \Delta \mathbf{d}. \end{aligned}$$

◇

O Teorema 2.1 é generalizado para deslocar um ponto sobre uma resolução inferior arbitrária de curva, $f^j, j < n$, pelo corolário a seguir.

Corolário 2.1 *Se $p = f^n(t) + \Delta \mathbf{d}$ é o ponto deslocado de $f^n(t)$ por $\Delta \mathbf{d}$, o mesmo efeito é obtido deslocando o ponto $f^j(t_j), 0 < j < n$, correspondente de $f^n(t)$, por $\Delta \mathbf{d}$.*

Prova: A função f^n pode ser expressa em função dos seus coeficientes de resoluções inferiores e wavelets como

$$f^n = \Phi^j C^j + \sum_{i=j}^{n-1} \Psi^i D^i.$$

Depois de deslocar o ponto $f^j(t_j)$ em $\Delta \mathbf{d}$, alguns coeficientes de C^j são modificados em ΔC^j . A expressão acima é reescrita como

$$\begin{aligned} \tilde{f}^n(t) &= \Phi^j(t) (C^j + \Delta C^j) + \sum_{i=j}^{n-1} \Psi^i D^i \\ &= \Phi^j(t) C^j + \Phi^j(t) \Delta C^j + \sum_{i=j}^{n-1} \Psi^i D^i \\ &= \Phi^j C^j + \sum_{i=j}^{n-1} \Psi^i D^i + \Delta \mathbf{d} \\ &= f^n(t) + \Delta \mathbf{d}. \end{aligned}$$

◇

A Figura 2.10 mostra os resultados do Teorema 2.1 e Corolário 2.1 depois do deslocamento do ponto $f^6(t) = (333.72, 506.37)$ por $\Delta \mathbf{d} = (-19.72, 64.62)$, onde a curva interior é representada por uma linha tracejada. O nível da curva mais refinada é $n = 6$. Quando um ponto correspondente de $f^6(t)$ sobre f^3 é deslocado em $\Delta \mathbf{d}$, a posição final $\tilde{f}^6(t)$, após a reconstrução, é a mesma (curva pontilhada externa da figura).

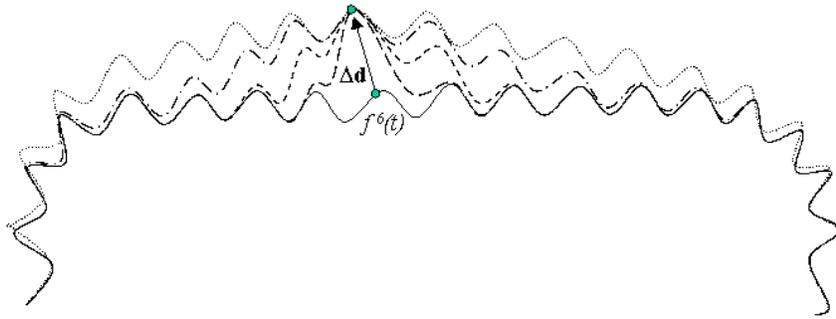


Figura 2.10: Vetor de deslocamento $\Delta \mathbf{d}$ aplicado em um ponto de cada resolução da curva, $f^j(t_j)$, $3 \leq j \leq 6$, por vez (curva pontilhada externa é resultado depois de deslocar um ponto $f^3(t_3)$; curva de linha tracejada interna é resultado após o deslocamento do ponto $f^6(t)$).

Uma outra consequência do Teorema 2.1 é que, se o vetor de deslocamento $\Delta \mathbf{d}$ é segmentado em vários vetores, $\Delta \mathbf{d}_1, \Delta \mathbf{d}_2, \dots, \Delta \mathbf{d}_m$, e cada vetor $\Delta \mathbf{d}_k$ é aplicado para manipulação direta sobre pontos respectivos de $f^n(t)$ em curvas de resoluções inferiores, f^j , então depois de reconstrução, o ponto $f^n(t)$ vê-se deslocado por $\Delta \mathbf{d}$. Este critério se formaliza pelo corolário a seguir.

Corolário 2.2 O deslocamento do ponto $f^n(t)$ por $\Delta \mathbf{d}$ é equivalente ao acumulado da seqüência de deslocamentos em $\Delta \mathbf{d}_i$ dos pontos relativos de $f^n(t)$ nas resoluções inferiores respectivas, onde $\Delta \mathbf{d}_1, \Delta \mathbf{d}_2, \dots, \Delta \mathbf{d}_m$ são segmentos de vetores de $\Delta \mathbf{d}$.

A prova deste corolário é imediata. Se se desloca um ponto $f^{n-i}(t_i)$, correspondente a $f^n(t)$, em $\Delta \mathbf{d}_i$ por vez em cada nível de resolução do contorno, o corolário 2.1 garante

que o ponto $f^n(t)$ é deslocado em $\Delta \mathbf{d}_i$. Desta forma, o acumulado dos deslocamentos de $f^n(t)$, finalmente, é $\Delta \mathbf{d} = \Delta \mathbf{d}_1 + \Delta \mathbf{d}_2 + \dots + \Delta \mathbf{d}_m$.

A Figura 2.11 mostra o efeito após a reconstrução quando o vetor de deslocamento $\Delta \mathbf{d}$ é segmentado por $\Delta \mathbf{d}_1, \Delta \mathbf{d}_2, \Delta \mathbf{d}_3$ and $\Delta \mathbf{d}_4$, e $f^6(t)$ é deslocado por $\Delta \mathbf{d}_1$, $f^5(t_5)$ por $\Delta \mathbf{d}_2$, $f^4(t_4)$ por $\Delta \mathbf{d}_4$ e $f^3(t_3)$ por $\Delta \mathbf{d}_3$.

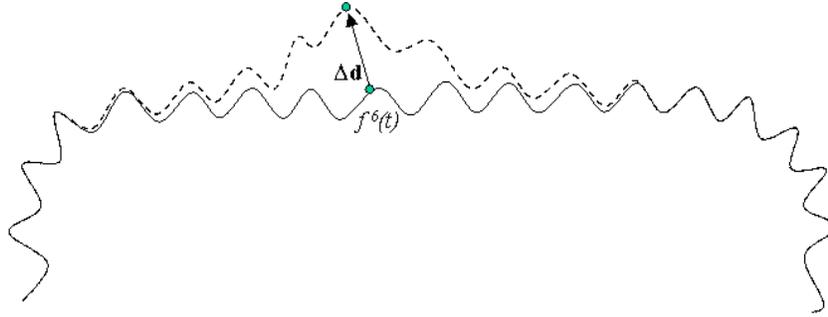


Figura 2.11: Vetor de deslocamento $\Delta \mathbf{d}$ segmentado aplicado sobre curvas de resolução inferior. A curva de linha tracejada é produzida depois da aplicação dos segmentos de vetores, $\Delta \mathbf{d}_1, \Delta \mathbf{d}_2, \dots, \Delta \mathbf{d}_m$, sobre cada curva de resolução inferior f^6, f^5, f^4 e f^3 , respectivamente.

2.4 Perturbações no contorno

O contorno de um corpo, expressa em seu nível refinado, f^n ou simplesmente f , pode ainda ser refinada para níveis mais superiores oscilando em torno de f^n . Esses contornos de níveis mais superiores vão definir as perturbações de f^n . Na verdade, em contornos de multiresolução, um nível de contorno passa a ser perturbação de seu equivalente de resolução imediatamente inferior. Com este critério, pode-se gerar a superfície de resolução superior, f^{n+1} ou f^{+1} , adicionando-se os coeficientes de detalhe com média zero.

Podem se criar níveis de resoluções superiores a n em forma controlada, sem intenção de adicionar níveis maiores que dois ou três, já que cada nível adicional requer o dobro do número de pontos de controle do nível imediatamente inferior, tal como visto na Seção 2.2.5. Assim, para gerar um contorno mais refinada do que f^n ,

ou seja f^{n+1} , requer-se 2^n pontos adicionais; do mesmo modo, para refinar f^{n+1} em um nível adicional, para f^{n+2} ou f^{+2} , são necessários 2^{n+2} pontos de controle. Este mecanismo de refinamento por inserção de pontos deve ser controlado, dependendo da complexidade das perturbações.

Podem ser criadas as perturbações sobre a superfície f^n de várias formas: reconstruindo contornos superiores a partir dos coeficientes de wavelets perturbados, perturbando o contorno por deformações locais, ou adicionando perturbações por bibliotecas. Nas duas primeiras formas de criar as perturbações, existem processos comuns: refinamento de pontos de controle por inserção e geração dos elementos perturbadores do contorno, enquanto o terceiro permite aproximar as perturbações por tolerâncias estatísticas.

Os elementos perturbadores são os vetores de detalhe ou deslocamento de pontos, segundo a forma de criação de rugosidade do contorno. Uma forma de gerar esses vetores é admitir que eles provêm de uma distribuição $N(0, \sigma^2)$, de maneira que o tamanho de maior predominância esteja dentro de um intervalo de confiança (λ) definido.

A geração das perturbações por reconstrução com coeficientes de wavelet perturbados consiste em reconstruir o contorno perturbado f^{n+1} a partir dos coeficientes C^n e os coeficientes wavelets D^n criados com uma certa tolerância estatística (ver Capítulo 6), já que formalmente não existem elementos wavelets do nível n . Sendo assim, o contorno perturbado é obtida como

$$f^{n+1} = f^n + \Psi^n D^n.$$

Na forma geral, quando são considerados m níveis adicionais de perturbação de f^n , os coeficientes de wavelet $D^r = (d_k^r)_{k=0, \dots, 2^{n+k}-1}$, para $r = n, \dots, n+m$, contribuem na definição das perturbações do contorno f^n como

$$f^{n+m} = f^n + \sum_{r=0}^m \Psi^{n+r} D^{n+r}. \quad (2.32)$$

Os coeficientes wavelet gerados no nível n e superiores podem ter uma distribuição não uniforme, descrevendo, desta forma, uma distribuição de rugosidade variada no contorno f^n . Com este critério, é mais conveniente para o usuário escolher uma forma de distribuição de perturbações em uma biblioteca de detalhes, onde exista distribuição constante, distribuição variada, etc. No Capítulo 6 se explica em detalhe a criação de rugosidades no contorno.

A Figura 2.12 ilustra como uma distribuição de perturbações é relacionada com os coeficientes wavelets de níveis n e $n+1$. O contorno f^{n+2} , que constitui a perturbação de f^{n+1} e de f^n , será obtida pela reconstrução, segundo (2.32), a partir dos coeficientes de wavelets D^n e D^{n+1} gerados.

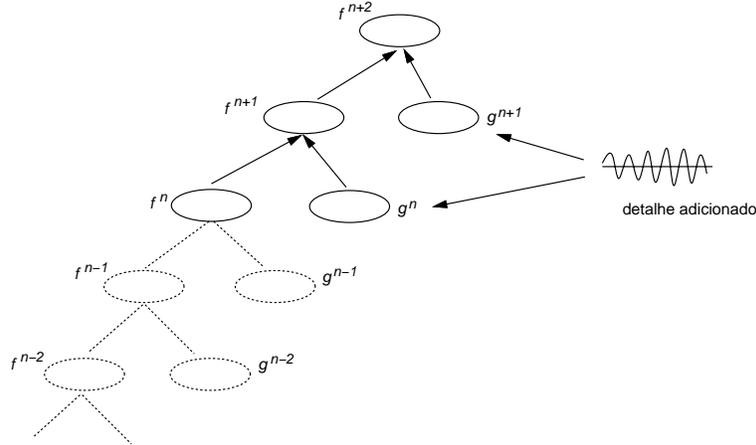


Figura 2.12: Criação das perturbações utilizando distribuição de coeficientes wavelets em dois níveis.

A geração de rugosidade via perturbação do contorno por deformações locais é uma abordagem inspirada na técnica de Szeliski e Terzopoulos [SzelTerz89], em que bombardeiam a superfície de terreno, no nível mais refinado, por perturbações geradas pela distribuição de Boltzmann com restrição variacional (minimizando a energia). Para gerar as rugosidades, neste trabalho, sobre f^n , pode-se usar a deformação local deslocando o ponto médio de cada segmento f_j^n por um vetor \mathbf{d}_j . O módulo desse vetor é gerado aleatoriamente segundo uma distribuição $N(0, \sigma^2)$, e sua direção sempre é normal à tangente do contorno no ponto a deslocar. Aplicado este processo a todos os segmentos do contorno, o mesmo processo pode ser feito com o contorno resultante, recursivamente.

Para o propósito de deformações locais no nível n , primeiro se refina f^n segundo os passos descritos na Seção 2.2.5, sendo f^{n+1} o contorno refinado. Cada segmento básico deste último contorno se deforma localmente, no ponto central de cada segmento, segundo o vetor de deslocamento dado pela biblioteca de perturbações. O vetor de deslocamento de cada ponto central do segmento básico será orientado perpendicularmente à tangente do segmento da curva nesse ponto. Se se desejar perturbações em mais níveis, sobre este contorno perturbado, será aplicado o mesmo processo que no

anterior: refinar, depois deformar localmente, cada segmento no ponto médio. Este processo é repetido m vezes, onde m é o nível de resoluções que se deseja perturbar. A Figura 2.13 ilustra a técnica descrita, para dois níveis de resolução de f^n .

A perturbação de f^n será, então, f^{n+m} , similar a (2.32). Nesta abordagem, se observa que é alterada a configuração de f^n , o que implica que após a decomposição via transformada wavelet do contorno f^{n+m} , os coeficientes wavelets vão ser diferentes de zero e a resolução \tilde{f}^n equivalente ao nível n será bastante aproximada à f^n original. Isto, também, vai implicar que as resoluções inferiores variem um pouco. Por esta razão, seria aconselhável redefinir o contorno f^n como sendo \tilde{f}^n após as perturbações.

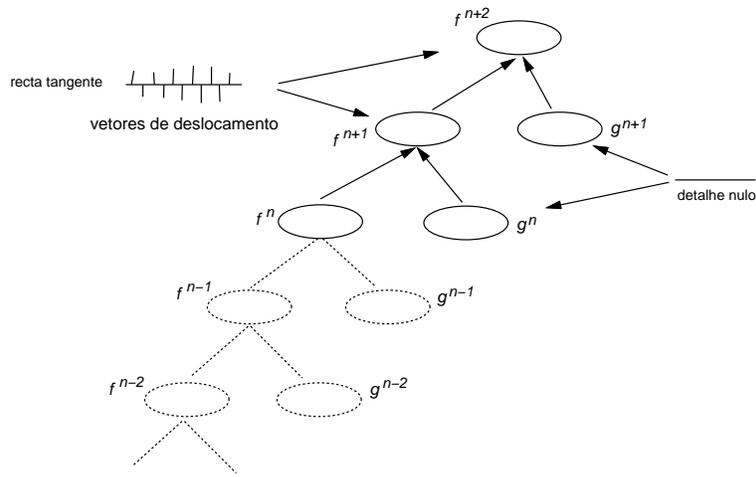


Figura 2.13: Criação das perturbações utilizando deformações locais em dois níveis.

Criando perturbações densas do contorno f^n , por qualquer das abordagens propostas acima, observa-se uma amostra de pontos de controle das versões superiores a f^n . Para evitar o uso excessivo de espaço de armazenamento, esses pontos são agrupados por tolerâncias estatísticas, detalhadas no Capítulo 5.

Capítulo 3

Construção de estruturas hierárquicas

A representação, em pré-processamento, de caixas hierárquicas envolvendo segmentos da superfície de um objeto faz com que o processo de comparação pela interseção de cada par de objetos em movimento ou em repouso seja eficiente e robusta. Embora se requera espaços adicionais para representar as estruturas hierárquicas de caixas envolventes, com o avanço da tecnologia, estas restrições passam a um segundo plano. O que interessa é a robustez e eficiência do algoritmo na detecção de contatos entre os corpos em interferência.

Neste capítulo formula-se uma abordagem de estruturas hierárquicas de caixas envolventes em multiresolução para objetos curvos de geometria arbitrária chamada “árvore de caixas envolventes orientadas de segmentos em multiresolução” (MOBB-tree). Para isto, se constroem as caixas orientadas envolventes baseadas em segmentos de superfícies curvilíneas, como uma extensão do método usado por Gottschalk, para objetos de geometria arbitrária formado por curvas. No próximo capítulo, complementa-se a geração das caixas orientadas envolventes considerando os detalhes de perturbação de objetos em multiresolução.

3.1 Estrutura hierárquicas

O OBBtree é uma árvore binária de caixas envolventes orientadas (*Oriented Bounding Boxes*, OBBs), de forma que, de um nível para outro, um pedaço da superfície do objeto coberto por uma caixa (pai) pode ser sub-divido em dois pedaços disjuntos para ser envolvidas por caixas pequenas cada pedaço (filhos). A Figura 3.1 dá uma idéia de como é uma árvore binária de OBBs.

Um OBB é uma caixa retangular cuja orientação depende do comportamento do conjunto de segmentos poliédricos (triangulares). O comportamento do conjunto de segmentos define a orientação de um OBB adaptado, isto faz com que esta estrutura seja mais eficiente do que a árvore esfera e a árvore de caixas de arestas alinhadas com os eixos coordenados (AABBs), porque os envelopes testados pela interseção são mais próximos aos pedaços de segmentos dos objetos em possível contato.

As técnicas de OBBtree são usadas em Ray-tracing e em outros cálculos de interferência [ArvoKirk89]. Em [RubiWhil80, WeHoGre84] os OBBs são usados em ray-tracing e modelagem hierárquica de objetos, sem a exigência de que as caixas sejam adaptadas e ajustadas à superfície. Gottschalk et al. aperfeçoaram o OBBtree para detecção de contatos em dinâmica de movimentos entre corpos rígidos de constituição poliédrica (malhas triangulares).

Neste trabalho, a criação de estruturas hierárquicas é inspirada na técnica de OBBtree, construindo-se os envelopes para segmentos de superfícies, chamados contornos, curvos em multiresolução e com detalhes de perturbação, dando origem a MOBBtree (*Multiresolution Oriented Bounding Box tree*). As OBBs de MOBBtree vão ser definidos pelas tendências de cada segmento curvo em multiresolução e suas respectivas perturbações. O processo de comparação pela interferência tem eficiência próxima à da árvore esfera, mas com maior precisão na determinação de segmentos de contorno de possível contato. O MOBBtree, na detecção de interferências, herda a eficiência e robustez da OBBtree reformulada por Gottschalk.

3.1.1 Estruturas MOBBtree

Na construção de uma estrutura MOBBtree são consideradas três componentes, não necessariamente sequenciais: primeiro, posicionamento de um adequado e ajustado

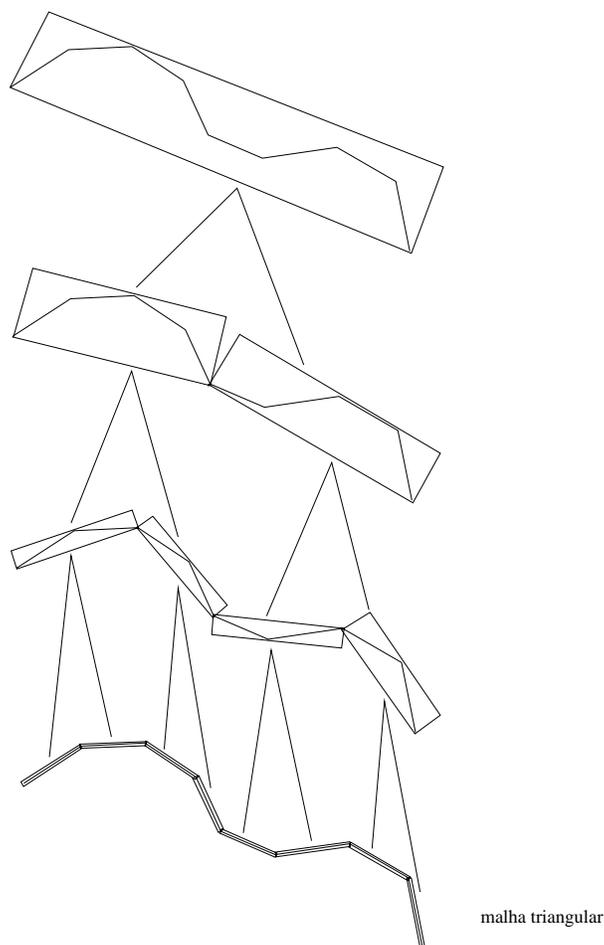


Figura 3.1: *Estrutura de um segmento de OBBtree.*

OBB, que contenha a coleção de segmentos de contorno irregular; segundo, aumento de caixas para incluir as perturbações dos segmentos respectivos; terceiro, agrupamento dos OBBs requeridos na estrutura hierárquica binária.

Em objetos definidos por malhas triangulares, os vértices dos triângulos formam parte da superfície do objeto, e tomando em consideração esse fato, construir caixas envolventes de características de um OBB é simples, tal como focado em [GoLiMa96]. Em objetos cujas fronteiras são definidas por pontos de controle que não necessariamente formam parte da superfície, a abordagem já não é tão simples, porque a mesma geometria de cada segmento da superfície deverá definir o OBB.

A concepção original de OBBtree para a geração dos descendentes (segmentos de polígonos envolvidos por um OBB pai é subdividido em dois subsegmentos disjuntos,

determinados pelo eixo maior do OBB pai, para se gerar os OBBs filhos) nem sempre funciona para segmentos côncavos, porque a subdivisão pela metade do segmento pode gerar mais de dois segmentos disjuntos não adjacentes. Para manter a robustez de OBBtree na criação de MOBBtree considera-se a união de dois segmentos adjacentes (filhos) para se gerar o segmento pai. Esta abordagem também demanda similar espaço do que OBBtree, só com uns processos adicionais na etapa de inclusão das perturbações nas caixas envolventes.

Uma caixa envolvente vai ser gerada a partir dos dados relativos às caixas inferiores, cuidando que os segmentos respectivos sejam contínuos. Outra característica observada nas caixas envolventes para curvas, é a necessidade do uso de técnicas de amostragens para discretizar os segmentos do contorno em elementos suficientemente significativos para definir os eixos de cada caixa, o uso de derivadas de primeiro e segundo ordem vão fornecer o máximo e mínimo relativo dos segmentos em direção dos eixos definidos.

Caixas envolventes

As caixas orientadas envolvendo segmentos de contorno do objeto são representadas por uma estrutura de árvore binária, de forma que a raiz é uma caixa contendo todo o objeto, cada um dos dois ramos contém uma parte do objeto relativo à caixa pai, etc. As folhas da árvore são as caixas que envolvem segmentos básicos do contorno do objeto. Neste sentido observam-se dois tipos de caixas: *caixa mínima* ou *básica* e *caixa superior*.

Uma caixa básica se caracteriza por envolver um único segmento básico e suas perturbações. Este tipo de caixa é gerada a partir do segmento básico e aumentada para incluir as perturbações respectivas. Uma caixa superior contém dois ou mais segmentos básicos e as perturbações relativas (na realidade uma caixa superior inclui as caixas básicas respectivas). Este tipo de caixa pode ser gerada de duas formas: a partir das caixas básicas adjacentes que vão ser incluídas, ou estimada a partir do segmento de resolução menor relativa (como se fosse uma caixa básica) depois aumentada incluindo as caixas básicas e as perturbações relativas. A Figura 3.2 mostra uma estrutura de MOBBtree com os dois tipos de caixas descritas.

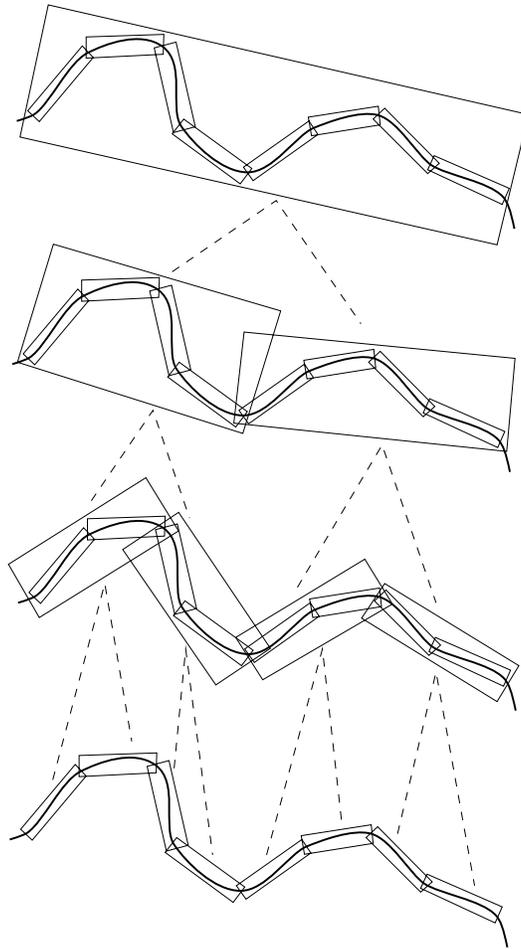


Figura 3.2: Hierarquia das caixas envolventes conservativas representadas em uma árvore binária.

3.2 Determinação da caixa envolvente adaptada

Um critério para construir uma caixa envolvendo um ou mais segmentos de curva consiste em considerar os pontos de controle relativos aos segmentos, já que seu fecho convexo envolve o segmento respectivo. Mas a caixa não ajustaria à curva adaptadamente, porque os pontos de controle nem sempre estão sobre a curva.

Uma outra forma de construir a caixa orientada consiste em determinar o retângulo de área mínima envolvendo os vértices das caixas mínimas. Para este processo, deve ser calculado o fecho convexo com esses pontos (o que requer um tempo $O(n \log n)$)

e depois usar um algoritmo eficiente para construir um retângulo de área mínima (em tempo $O(n)$) [FreeSha75]. Este método consegue construir caixas ajustada para cobrir o segmento, mas requer um tempo total $O(n \log n)$.

A matriz de covariância destes pontos, calculada em tempo linear, vai permitir determinar os dois eixos perpendiculares, os quais determinam a orientação da caixa. Isto é, as arestas da caixa retangular serão paralelas aos eixos. Os eixos e a média dos pontos, como centroide temporário, vão permitir calcular as dimensões da caixa mais ajustada às caixas mínimas. A partir da caixa ajustada será recalculado seu centroide verdadeiro.

3.2.1 Uso da matriz de covariância

Com n pontos \mathbf{p}_i relativos aos segmentos do contorno em análise, é determinada a matriz de covariância σ , que tem a forma

$$\sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix} \quad (3.1)$$

cujos elementos são calculados tradicionalmente como

$$\sigma_{xy} = \frac{1}{n} \sum_{i=1}^n (p_i^x - \mu^x)(p_i^y - \mu^y), \quad \text{para } \mu = \frac{1}{n} \sum_{j=1}^n \mathbf{p}_j \quad \text{e } 1 \leq x, y \leq 2, \quad (3.2)$$

mas pode variar dependendo do caso a se resolver, como se verá posteriormente.

A matriz de covariância permite calcular a dispersão dos dados em qualquer direção. Isto ajuda a determinar a tendência da agrupação dos dados em forma elíptica [Morrison76, JohnWich92], onde os eixos são determinados pela orientação dos autovetores da matriz de covariância e o tamanho dos eixos pelos autovalores.

A matriz de covariância, neste caso 2×2 , permite determinar 2 autovetores e dois autovalores (cada vetor com um único autovalor, mas não necessariamente o contrário [JohnWich92]). Desta forma, com o maior autovalor será calculado o eixo de maior lado da caixa, e com o segundo autovalor será calculado o segundo eixo correspondente ao lado menor da caixa.

Os autovetores de σ são ortogonais e os autovalores respectivos são reais e positivos, desde que a matriz de covariância σ seja simétrica e definida positiva [Morrison76].

Assim, os autovetores normalizados vão constituir uma base para gerar o espaço de distribuição dos pontos.

3.2.2 Determinação dos eixos das caixas envolventes

Dada uma matriz simétrica \mathbf{A} de ordem $p \times p$, e o vetor \mathbf{X} de p elementos. Então \mathbf{X} é um autovetor de \mathbf{A} correspondendo a um valor real λ chamado autovalor, se

$$\mathbf{A}\mathbf{X} = \lambda\mathbf{X}, \quad \text{onde } \mathbf{X} \text{ é não nulo.} \quad (3.3)$$

Para o caso bidimensional, não será necessário usar um algoritmo geral de cálculo de autovalores e autovetores, porque sua dedução pode se determinar facilmente.

Se a matriz simétrica 2×2 é dada por

$$\mathbf{A} = \begin{pmatrix} a & b \\ b & c \end{pmatrix},$$

então a expressão (3.3) pode ser reformulada como

$$(\mathbf{A} - I\lambda)\mathbf{X} = \mathbf{0}, \quad (3.4)$$

que fornece os dois autovalores da forma

$$\lambda_1 = \frac{a + c + \sqrt{(a - c)^2 + 4b^2}}{2},$$

$$\lambda_2 = \frac{a + c - \sqrt{(a - c)^2 + 4b^2}}{2}.$$

Como existem dois autovetores da matriz \mathbf{A} , cada um é associado a um autovalor, da forma

$$\mathbf{X}_1 = (-b, a - \lambda_1)$$

$$\mathbf{X}_2 = (-b, a - \lambda_2).$$

Os dois autovetores são ortogonais, porque

$$\begin{aligned} \mathbf{X}_1 \cdot \mathbf{X}_2 &= (-b, a - \lambda_1)(-b, a - \lambda_2) \\ &= b^2 + a^2 - (\lambda_1 + \lambda_2)a + \lambda_1\lambda_2 \\ &= b^2 + a^2 - \left[\frac{a + c + \sqrt{(a - c)^2 + 4b^2}}{2} + \frac{a + c - \sqrt{(a - c)^2 + 4b^2}}{2} \right] a + \end{aligned}$$

$$= 0.$$

$$\left(\frac{a + c + \sqrt{(a - c)^2 + 4b^2}}{2} \right) \left(\frac{a + c - \sqrt{(a - c)^2 + 4b^2}}{2} \right)$$

Existem casos especiais que são levados em consideração no cálculo de autovetores. Por exemplo, quando os elementos da matriz, $b = 0$ e, ou a ou c é diferente de zero (quando os pontos que determinam a covariância formam uma reta vertical ou horizontal). Neste caso, o autovetor nulo será posto como vetor perpendicular ao outro autovetor diferente de nulo. No caso os autovalores λ_1 e λ_2 sejam iguais (simetria de pontos), definem autovetores paralelos ou nulos; neste caso serão adotados como soluções as direções horizontal e vertical.

Em geral, os autovalores e autovetores, em termos dos elementos da matriz de covariância, são calculados como:

$$\lambda = \frac{\sigma_{11} + \sigma_{22} \pm \sqrt{(\sigma_{11} - \sigma_{22})^2 + 4(\sigma_{12})^2}}{2},$$

$$\mathbf{X}_i = (-\sigma_{12}, \sigma_{11} - \lambda_i), \quad i = 1, 2.$$

3.3 Caixa básica

Uma caixa básica deve envolver, em forma adaptada e ajustadamente, um segmento f_j e suas respectivas perturbações. Talvez esta caixa não seja de área mínima mas deve ser adaptada à tendência do segmento e incluindo-o ajustadamente junto com suas perturbações. Nesse sentido, o primeiro passo fundamental é determinar os dois eixos perpendiculares que expressam a orientação da caixa. Depois, então, se determinará a caixa ajustada ao segmento, para finalmente incorporar as suas perturbações.

3.3.1 Determinação dos eixos da caixa envolvendo um segmento de curva

Quatro pontos de controle podem definir um segmento básico de curva de uma das formas: segmento cúbico, segmento parabólico, segmento linear ou um ponto (quando o comprimento do segmento é mínimo). A Figura 3.3 mostra estes possíveis casos.

Estas considerações vão permitir a escolha de cálculo dos eixos para uma caixa envolvendo o segmento de curva.

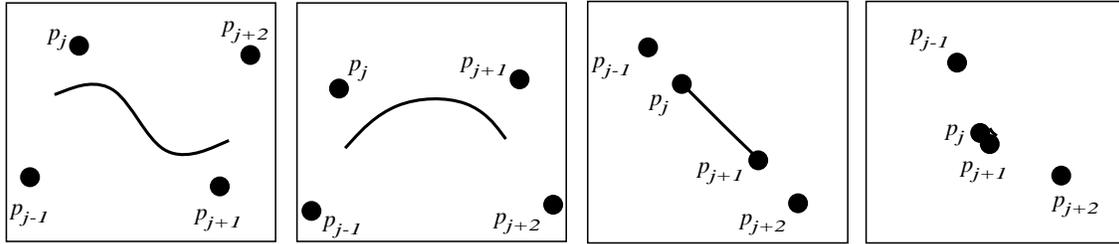


Figura 3.3: Casos de segmentos de curva f_j e seus pontos de controle respectivos.

Uma alternativa é considerar que um dos eixos, e_1 , da caixa tem a direção do vetor $\mathbf{f}_j(1) - \mathbf{f}_j(0)$. O segundo eixo e_2 seria o vetor unitário perpendicular a e_1 . Neste método nada garante que a orientação e as dimensões da caixa sejam adaptadas ao segmento f_j , porque em segmentos com um ponto de inflexão o vetor que une os extremos de f_j pode ser próximo à diagonal de uma suposta caixa mais adaptada que a calculada (vide Figura 3.4). Segmentos de curva sem pontos de inflexão podem usar adequadamente este método, mas quando o segmento é fechado, o método deixa de ser adequado. De qualquer forma, um segmento f_j , neste trabalho, nunca é fechado.

A segunda forma de determinar os eixos da caixa usa a tendência dos pontos ao longo do segmento da curva (o melhor ajuste). Para isto, deve-se amostrar n pontos, $\{\mathbf{p}_k\}_{k=1,\dots,n}$, regularmente distribuídos sobre o segmento da curva, incluindo os extremos (vide Figura 3.5). Com estes n pontos amostrados se computa a matriz de covariância de pontos. A seguir se determinam os dois autovetores da matriz de covariância. Como nas seções anteriores, a tendência da distribuição dos pontos da amostra determina a direção dos autovetores. Estes autovetores serão os eixos da caixa orientada e adaptada ao segmento de curva.

Considerando a técnica de amostragem, o elemento da fila x e coluna y da matriz da covariância para n pontos amostrados é dado por

$$\sigma_{xy} = \frac{1}{n-1} \sum_{k=1}^n (p_k^x - \bar{p}^x)(p_k^y - \bar{p}^y), \quad (3.5)$$

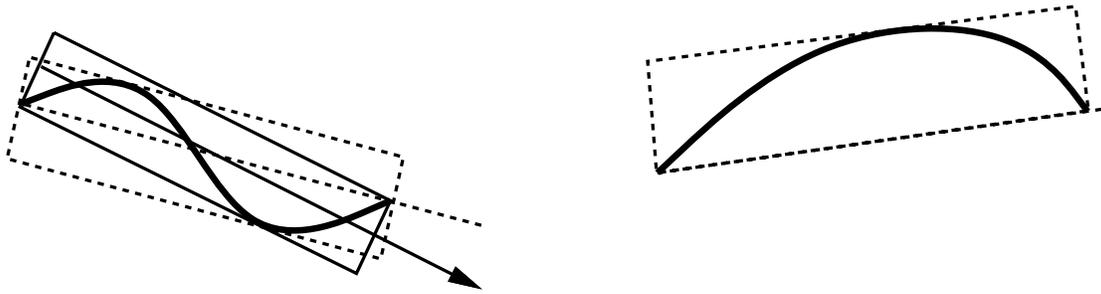


Figura 3.4: *Caixas envolvendo um segmento de curva f_j : caixa de linha pontilhada relativo ao eixo determinado por extremos do segmento; caixa linha contínua relativo ao eixo determinado com melhor ajuste.*

onde $\bar{\mathbf{p}} = \sum_{k=1}^n \bar{\mathbf{p}}_k$ é a média dos pontos amostrados.

O valor de n depende do comprimento l_j do segmento f_j . Para isto, estabelece-se um comprimento mínimo (l_{min}) dos segmentos, de modo que se l_j for maior que l_{min} , a amostra é composta por 5 pontos ($n = 5$) regularmente distribuídos no segmento f_j , incluindo os seus extremos. Caso contrário, a amostra é unicamente composta pelos pontos extremos do segmento ($n = 2$). Este número de amostras permite que a caixa básica seja elaborada em tempo constante

O comprimento de arco l_j de um segmento de curva $\mathbf{f}(t)$, com $a \leq t \leq b$, é dado por

$$l_j = \int_a^b |\mathbf{f}'(t)| dt. \quad (3.6)$$

Em vista que o segmento de curva é $\mathbf{f}_j(t)$ para $0 \leq t \leq 1$, a função $|\mathbf{f}'_j(t)|$ é a ser raiz quadrada de uma função de ordem 4, sendo da forma

$$\begin{aligned} |\mathbf{f}'_j(t)| &= \frac{1}{2} |(\mathbf{v} t^2 + \mathbf{w} t + \mathbf{r})| \\ &= \frac{1}{2} \sqrt{|\mathbf{v}|^2 t^4 + 2\mathbf{v} \cdot \mathbf{w} t^3 + (2\mathbf{v} \cdot \mathbf{r} + |\mathbf{w}|^2) t^2 + 2\mathbf{w} \cdot \mathbf{r} t + |\mathbf{r}|^2}, \end{aligned} \quad (3.7)$$

onde os vetores \mathbf{v} , \mathbf{w} e \mathbf{r} são expressos em termos dos pontos de controle relativos ao

segmento f_j (como formulados na Seção 2.1), como segue:

$$\left. \begin{aligned} \mathbf{v} &= p_{j+2} - 3p_{j+1} + 3p_j - p_{j-1} \\ \mathbf{w} &= 2p_{j+1} - 4p_j + 2p_{j-1} \\ \mathbf{r} &= p_{j+1} - p_{j-1} \end{aligned} \right\}. \quad (3.8)$$

Para efeitos de expansão e operação, considera-se cada vetor \mathbf{v} , \mathbf{w} e \mathbf{r} em seus componentes básicos. Assim,

$$\begin{aligned} |(\mathbf{v} t^2 + \mathbf{w} t + \mathbf{r})| &= (v_x t^2 + w_x t + r_x)^2 + (v_y t^2 + w_y t + r_y)^2 \\ &= (v_x^2 + v_y^2)t^4 + 2(v_x w_x + v_y w_y)t^3 + (2(v_x r_x + v_y r_y) + w_x^2 + w_y^2)t^2 \\ &\quad + 2(w_x r_x + w_y r_y)t + r_x^2 + r_y^2 \\ &= |\mathbf{v}|^2 t^4 + 2\mathbf{v} \cdot \mathbf{w} t^3 + (2\mathbf{v} \cdot \mathbf{r} + |\mathbf{w}|^2)t^2 + 2\mathbf{w} \cdot \mathbf{r} t + |\mathbf{r}|^2, \end{aligned}$$

A integral expressa em (3.6) para a função (3.7) não tem solução analítica, por isso se usará integral aproximada. A quadratura de Gauss [ChaCan85, PrTeVe+92] fornece uma solução quase exata, considerando 4 pontos de avaliação, para o propósito deste trabalho.

O comprimento de arco l_j deve ser armazenado na estrutura de dados que descreve a caixa envolvente B_j do segmento f_j . No cálculo de caixas superiores envolvendo mais de uma caixa inferior, é usado l_j como peso de cada caixa básica.

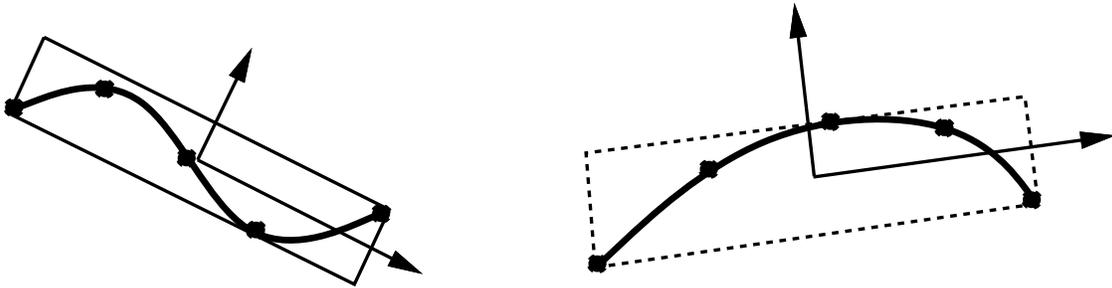


Figura 3.5: Eixo calculado por melhor ajuste com n pontos amostrados.

3.3.2 Caixa ajustada a um segmento de curva

Dados os eixos unitários \mathbf{e}_1 e \mathbf{e}_2 com origem no ponto médio $\bar{\mathbf{p}}$ dos n pontos amostrados no segmento f_j , os lados da caixa são determinados procurando os pontos sobre

f_j mais afastados das restas definidas pelos eixos.

Os pontos mais afastados de um eixo e_x , $x = 1, 2$, são determinados de forma que os vetores tangentes do segmento f_j nesses pontos sejam paralelos ao eixo e_x , ou perpendicular ao outro eixo e_y . Isto é, para determinar os pontos afastados do eixo e_x , devem-se calcular os pontos $\mathbf{f}_j(t)$ tal que

$$\mathbf{f}'_j(t) \cdot \mathbf{e}_y = 0. \quad (3.9)$$

Expandindo a expressão acima, considerando que a derivada de um segmento f_j é de segunda ordem, tem-se

$$\begin{aligned} \mathbf{f}'_j(t) \cdot \mathbf{e}_y &= \frac{1}{2}(\mathbf{v} t^2 + \mathbf{w} t + \mathbf{r}) \cdot \mathbf{e}_y \\ &= \frac{1}{2}(\mathbf{v} \cdot \mathbf{e}_y) t^2 + (\mathbf{w} \cdot \mathbf{e}_y) t + \mathbf{r} \cdot \mathbf{e}_y. \end{aligned}$$

Resolvendo a equação de segundo grau, tem-se os valores de t ,

$$t = \frac{-(\mathbf{w} \cdot \mathbf{e}_y) \pm \sqrt{(\mathbf{w} \cdot \mathbf{e}_y)^2 - 4(\mathbf{v} \cdot \mathbf{e}_y)(\mathbf{r} \cdot \mathbf{e}_y)}}{2(\mathbf{v} \cdot \mathbf{e}_y)}. \quad (3.10)$$

Os coeficientes \mathbf{v} , \mathbf{w} e \mathbf{r} são derivados da expressão da derivada de curvas B-splines cúbicas, tal como em (3.8).

Os valores $0 \leq t \leq 1$ em (3.10) determinam os pontos pelos quais as arestas paralelas da caixa vão passar. Não obstante, teriam que se considerar os casos em que t pode ter um valor, dois valores, infinitos valores, ou não exista solução. As interpretações para estes casos é que, quando existem infinitas soluções (para $\mathbf{v} \cdot \mathbf{e}_y = 0$), o segmento é um pedaço de reta perpendicular ao eixo e_y . Neste caso, não existem pontos mais afastados do eixo e_x , mas pode se considerar virtualmente como existindo pontos afastados minimamente para ambos os lados do segmento. No caso que não exista solução, os extremos do segmento são os pontos mais afastados do eixo em análise. No caso de haver uma solução, o ponto determinado é o mais afastado e um dos extremos do segmento é o segundo ponto mais afastado. Finalmente, quando existem duas soluções, esses dois pontos são os pontos mais afastados do eixo e_x .

Dois pontos mais afastados em relação ao eixo e_1 determinam o comprimento de lado da caixa em direção do eixo e_2 , e no outro caso o segundo lado da caixa. O centróide da caixa que envolve um segmento de curva é recalculado a partir da caixa

determinada já que o centróide dado pela média das amostras não necessariamente indica o centróide da caixa.

A caixa básica definida da forma explicada até aqui deve incluir também as perturbações do respectivo segmento de contorno, para o qual deverá se aumentar a caixa para os lados em uma tolerância definida pelas perturbações. A definição de tolerância discutem-se nos próximos capítulos.

Os elementos armazenados na estrutura de dados definindo uma caixa básica devem ser: centroide, os eixos, comprimento dos lados, comprimento do segmento envolvido, e a tolerância associada às perturbações do segmento.

3.4 Caixa superior a partir das caixas básicas

Nesta seção se formaliza a construção de uma caixa superior a partir de caixas básicas adjacentes, deixando para o próximo capítulo a construção da caixa superior em multiresolução devido a que se requer, primeiramente, uma definição de perturbações em multiresolução.

Para se construir uma caixa superior B , se requer n segmentos de curva adjacentes, limitados por caixas mínimas relativas a duas caixas superiores adjacentes que vão ser filhos de B , para se estimar a dispersão destes segmentos. Na estrutura das caixas estão registrados os dados relativos a seus vértices, seu centróide e o comprimento de arco do segmento que envolve.

A matriz de covariância dos centróides das caixas mínimas que envolvem segmentos do contorno fornece uma medida de sua dispersão. Para melhores resultados, cada centróide deve ser tomado com peso, proporcional à importância da caixa correspondente. Podem ser adotados, por exemplo, a área da caixa ou o comprimento do segmento correspondente. Desta forma evita-se a influência de uma concentração de segmentos pequenos em um setor do contorno, para uma orientação da caixa mais adaptada aos segmentos.

Seja \mathbf{p}_i o centróide da caixa mínima B_i , e seja l_i o comprimento de arco de cada segmento básico. A média dos n segmentos é dado por

$$\boldsymbol{\mu} = \frac{1}{l} \sum_{i=1}^n l_i \mathbf{p}_i, \quad (3.11)$$

onde

$$l = \sum_{i=1}^n l_i.$$

O elemento σ_{xy} , que é o elemento da linha x e coluna y da matriz de covariância $\boldsymbol{\sigma}$, é calculado, neste caso, como

$$\sigma_{xy} = \frac{1}{l} \sum_{i=1}^n l^i (p_i^x - \mu^x)(p_i^y - \mu^y), \quad \text{para } 1 \leq x, y \leq 2. \quad (3.12)$$

A matriz de covariância desses n pontos é

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix}, \quad (3.13)$$

cujo cálculo é linear para n dados.

Com a matriz de covariância são facilmente calculados os eixos da caixa desejada, com os procedimentos dados na Seção 3.2.

Determinação da caixa superior ajustada ao contorno

Dados os eixos \mathbf{e}_1 e \mathbf{e}_2 , se calcula o lado paralelo ao eixo \mathbf{e}_1 e o lado paralelo ao eixo \mathbf{e}_2 da nova caixa B .

Se considera que o ponto médio $\boldsymbol{\mu}$ nem sempre vai constituir o centróide de B , mas se usará como origem das retas perpendiculares geradas por \mathbf{e}_1 e \mathbf{e}_2 , sobre os quais se projetarão os vértices das caixas mínimas, para determinar a distância dos pontos mais afastados da respectiva reta.

Dados os vértices $\{\mathbf{v}_i\}_{i=1, \dots, 4n}$ de n caixas mínimas, as distâncias dos pontos mais afastados da reta gerada por \mathbf{e}_x , para ambos os lados, são determinadas por

$$l_x^- = \min \{l_x^i\}_{i=1, \dots, 4n} \text{ e } l_x^+ = \max \{l_x^i\}_{i=1, \dots, 4n}$$

onde

$$l_x^i = (\mathbf{v}_i - \boldsymbol{\mu}) \cdot \mathbf{e}_x, \quad \text{para } i = 1, \dots, 4n.$$

O lado l_x de B é calculado como $l_x = l_x^+ - l_x^-$. Para os dois eixos, variando $x = 1, 2$, são determinados os dois lados de B , com os quais fica definida a nova caixa. Os quatro vértices de B são calculados combinando os quatro vetores que saim da

origem μ , $l_1^+ \cdot \mathbf{e}_1$, $l_1^- \cdot \mathbf{e}_1$, $l_2^+ \cdot \mathbf{e}_2$ e $l_2^- \cdot \mathbf{e}_2$. Com estes elementos se recalculam o centróide exato da caixa B .

Como se verá, a definição de uma caixa ajustada ao segmento do contorno composto por n segmentos básicos é feita em tempo linear $O(n)$, e a determinação dos eixos também é linear, por isto uma caixa superior é feita em tempo linear.

3.5 Árvore binária de caixas envolventes

O método para a representação hierárquica das caixas, usado neste trabalho, é conhecido como método de geração da árvore em forma *Bottom-Up*. Pode-se considerar o método *Top-Down*, mas seria difícil controlar a adjacência dos segmentos do contorno de um objeto complexo.

Uma vez construídas as caixas básicas, como descritos na seção anterior, são construídas as caixas superiores com base nas caixas básicas, de forma que cada caixa superior contenha elementos contidos nas duas caixas inferiores, mas não as duas caixas. Desta forma, se constrói, em forma *Bottom-Up*, a árvore. A Figura 3.6 ilustra oito níveis de caixas envolvendo segmentos de um objeto com 256 pontos de controle.

Os elementos envolvidos por caixas superiores, vistas até este seção, são os segmentos básico do contorno. Existem outros elementos que definem a característica complexa do contorno que devem ser incluídos na caixa, tais como perturbações do segmento de curvas básicas, os que são tratados no próximo capítulo. Mas a filosofia de construção da árvore binária é única.

A estrutura para representar a árvore de caixas orientadas deve ser capaz de indicar a raiz da árvore, para comparação por interferência em forma hierárquica, e também permitir percorrer caixas de um dado nível, para verificação do estado de caixas adjacentes.

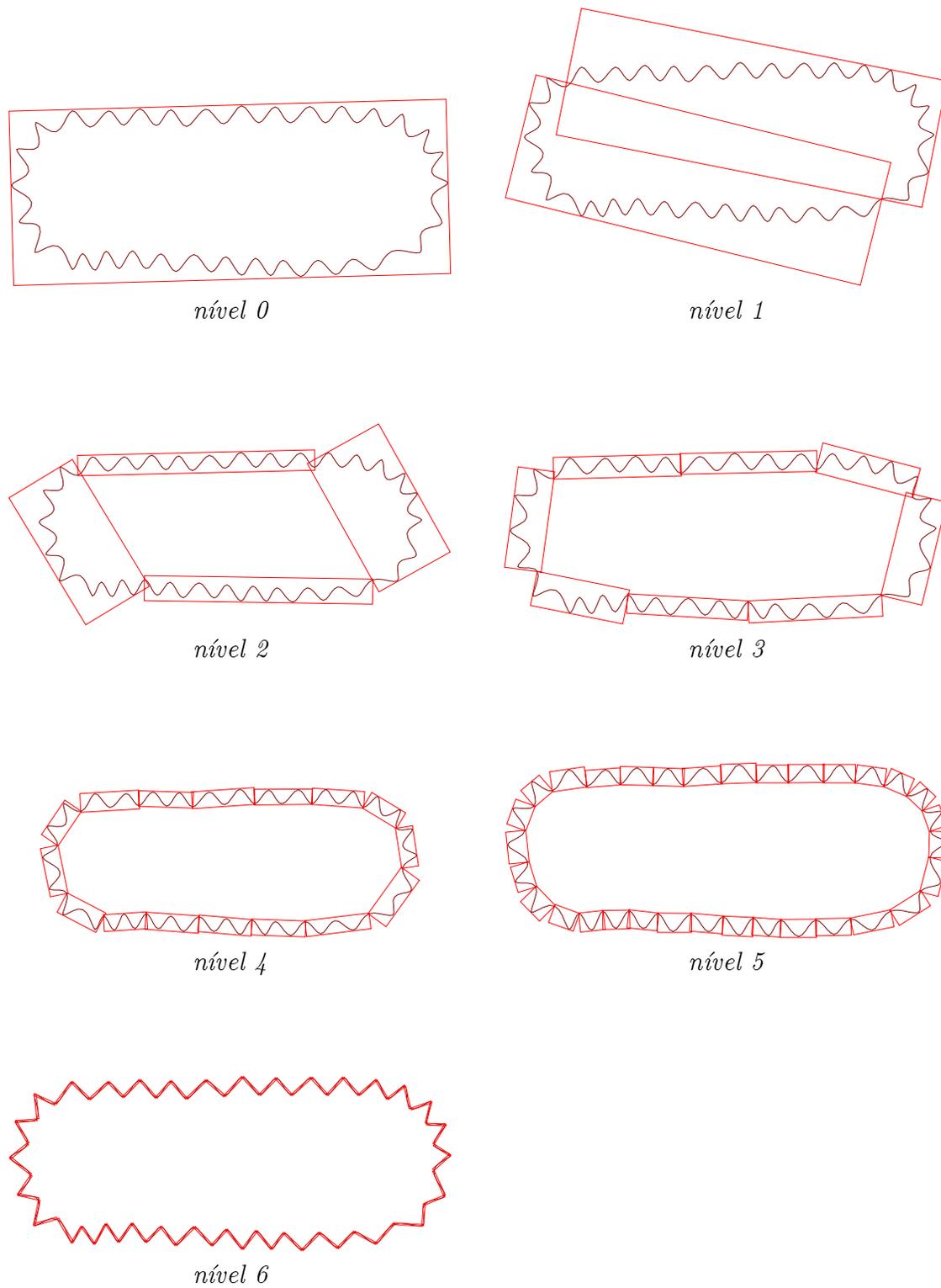


Figura 3.6: Caixas orientadas hierárquicas envolvendo segmentos do contorno de um objeto de 64 pontos de controle.

Capítulo 4

Análise de perturbações

Os contornos dos objetos definidos no Capítulo 2 podem apresentar perturbações irregulares como produto da distribuição não uniforme de seus pontos de controle, de tal forma a definir os detalhes que são comuns nos objetos do mundo real. Com este critério, Zorin et al. [ZoSchSw97] abordam a representação de objetos de geometria complexa em 3D, usando malhas triangulares em multiresolução baseada no esquema de subdivisão de Loop [Loop94], combinado com o algoritmo de suavização de Taubin [Taubin95]. Zorin et al. usam a representação de malha poligonal em que os pontos de controle são os vértices dos elementos da malha, e os detalhes são dados pela diferença dos vértices de um nível refinado para seu nível inferior. Esses detalhes são tratados como vetores orientados em relação a seus sistemas de coordenadas locais. As operações nos objetos em multiresolução, são dadas sobre os pontos de controle (vértices), ficando os vetores de detalhe inalteráveis. A orientação em coordenada local de cada vetor de detalhe é armazenada, no processo de análise, nas estruturas relativas aos vértices. No processo de síntese são recomputados os vetores de detalhe de coordenada local ao sistema do objeto, para serem adicionados nos vértices correspondentes.

Critério análogo à abordado por Zorin et al. é focado neste trabalho, para a operação de detalhes que vão definir as perturbações dos objetos, mas considerando que os detalhes sobre uma superfície se dão a nível de curvas e não a nível de seus pontos de controle, porque eles não necessariamente estão sobre o contorno em análise.

Neste capítulo, é analisado o comportamento dos coeficientes de detalhe que defi-

nem as curvas de diferença entre um nível dado e seu inferior. Esses elementos serão utilizados para definir os elementos de perturbação da superfície. Finalmente, enfoca-se a construção de caixas envolventes dos elementos da superfície em multiresolução, por um processo de aumento de caixa.

4.1 Detalhes do contorno em multiresolução

Considere um contorno f^j , com coeficientes de escala C^j e detalhe D^j . Aplicando o processo de reconstrução de nível j para sua representação no nível $j + 1$, sem considerar a adição de detalhes (como se fosse um processo de refinamento, Seção 2.2.5), o contorno resultante de nível $j + 1$ (\bar{f}^{j+1}) é o mesmo contorno de j . O contorno obtido é a projeção de f^j sobre o espaço de nível $j + 1$,

$$\bar{f}^{j+1} = \text{proj}_{j+1} f^j,$$

o que é equivalente a se estabelecer

$$\bar{f}^{j+1} \equiv f^j.$$

A diferença entre as duas representações de um mesmo contorno é que \bar{f}^{j+1} tem o mesmo número de pontos de controle que o contorno refinado f^{j+1} , enquanto f^j tem menor número de pontos de controle. Assim, os segmentos que faltam a f^j para ser f^{j+1} são os detalhes do nível j reconstruídos (\bar{g}^{j+1}), o que simplesmente pode ser a diferença aritmética entre estes dois contornos, segundo o método seguido em [ZoSchSw97], já que tem-se os mesmos elementos de controle em ambos os níveis, e ambos os contornos estão no mesmo espaço para se operar. Então a diferença aritmética entre a curva do nível $j + 1$ e seu equivalente no nível j é

$$\bar{g}^{j+1} = f^{j+1} - \bar{f}^{j+1},$$

a qual é equivalente à operação de projeção da curva de detalhe g^j obtida com os elementos de D^j , como

$$\bar{g}^{j+1} = \text{proj}_{j+1} g^j.$$

É necessário que os elementos de ambos os níveis estejam no mesmo espaço para serem operados, já que não é possível fazer operações aritméticas ou comparações com elementos de níveis diferentes.

Uma vez que os detalhes estão em relação à versão refinada, $j + 1$, é possível recalculá-las suas coordenadas locais em relação a cada ponto de controle. A orientação desses vetores de detalhe permitirão computar a magnitude de perturbação do contorno f^j , vide Figura 4.1.

Seguindo as deduções de convolução, cada coeficiente C^{j+1} é gerado a partir dos coeficientes C^j e D^j , de forma geral (2.29), como

$$\begin{aligned} c_r^{j+1} &= \sum_k h_{r-2k} c_r^j + \sum_s g_{r-2s} d_r^j \\ &= \bar{c}_r^{j+1} + \bar{d}_r^{j+1} \end{aligned} \quad (4.1)$$

A função f^{j+1} é imediatamente obtida a partir dos coeficientes em (4.1). Os 2^{j+1} segmentos de curva são obtidos, para o caso de bases B-splines cúbicas, de acordo com (2.4). Então, um segmento r de uma curva do nível $j + 1$ é dado por

$$\begin{aligned} f_r^{j+1}(t) &= \tilde{N}(t) \begin{bmatrix} c_{r-1}^{j+1} \\ c_r^{j+1} \\ c_{r+1}^{j+1} \\ c_{r+2}^{j+1} \end{bmatrix} \\ &= \tilde{N}(t) \begin{bmatrix} \bar{c}_{r-1}^{j+1} + \bar{d}_{r-1}^{j+1} \\ \bar{c}_r^{j+1} + \bar{d}_r^{j+1} \\ \bar{c}_{r+1}^{j+1} + \bar{d}_{r+1}^{j+1} \\ \bar{c}_{r+2}^{j+1} + \bar{d}_{r+2}^{j+1} \end{bmatrix} \\ &= \tilde{N}(t) \begin{bmatrix} \bar{c}_{r-1}^{j+1} \\ \bar{c}_r^{j+1} \\ \bar{c}_{r+1}^{j+1} \\ \bar{c}_{r+2}^{j+1} \end{bmatrix} + \tilde{N}(t) \begin{bmatrix} \bar{d}_{r-1}^{j+1} \\ \bar{d}_r^{j+1} \\ \bar{d}_{r+1}^{j+1} \\ \bar{d}_{r+2}^{j+1} \end{bmatrix} \\ &= \bar{f}_r^{j+1}(t) + \bar{g}_r^{j+1}(t). \end{aligned} \quad (4.2)$$

Com os resultados de (4.2), pode-se, claramente, ver que a função diferença entre os níveis $j + 1$ e j obtém-se diretamente dos detalhes convoluídos com os detalhes de

nível j . Dai que a função de detalhe do nível j é dada por

$$\bar{g}_r^{j+1}(t) = \tilde{N}(t) \begin{bmatrix} \bar{d}_{r-1}^{j+1} \\ \bar{d}_r^{j+1} \\ \bar{d}_{r+1}^{j+1} \\ \bar{d}_{r+2}^{j+1} \end{bmatrix} \quad (4.3)$$

A expressão (4.2), cujo segundo termo pode ser calculado como (4.3), é equivalente a se formular

$$\bar{g}_r^{j+1}(t) = f_r^{j+1}(t) - \bar{f}_r^{j+1}(t). \quad (4.4)$$

Em vista de que as funções f^j e \bar{f}^{j+1} se referem a um mesmo contorno, onde f^j é dado por 2^j segmentos de curva, enquanto \bar{f}^{j+1} é dada por 2^{j+1} segmentos, a correspondência entre os elementos destes dois níveis é dado, como explicado na Seção 2.3, onde se introduziu o critério de convergência entre os elementos de dois níveis de resoluções adjacentes (j e $j + 1$), como

$$f_k^j \xrightarrow{\text{síntese}} f_{2k}^{j+1} \cup f_{2k+1}^{j+1}. \quad (4.5)$$

Os pontos de controle relativos a estes segmentos são correspondentes.

O processo de síntese sem considerar os detalhes, para se obter as expressões acima deduzidas, será chamado de *síntese-escala*, onde os pontos de controle também são correspondentes.

$$f_k^j \xrightarrow{\text{síntese-escala}} \bar{f}_{2k}^{j+1} \cup \bar{f}_{2k+1}^{j+1}. \quad (4.6)$$

Mesmo critério é considerado para os casos de detalhe entre esses dois níveis, cujo processo será chamado *síntese-detalle*.

$$g_k^j \xrightarrow{\text{síntese-detalle}} \bar{g}_{2k}^{j+1} \cup \bar{g}_{2k+1}^{j+1}. \quad (4.7)$$

Com os critérios seguidos até aqui, pode-se dizer que as perturbações sobre o contorno f^j são dadas pelo comportamento da função g^j , que é equivalente à expressão

$$g_k^j = \bar{g}_{2k}^{j+1} \cup \bar{g}_{2k+1}^{j+1}. \quad (4.8)$$

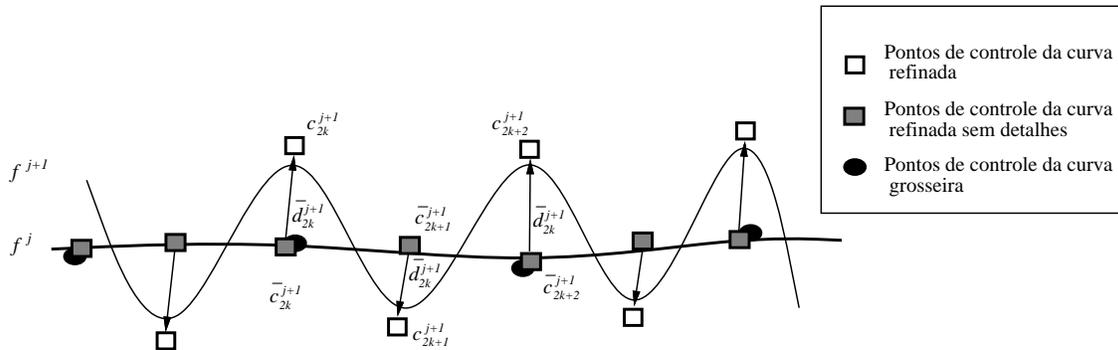


Figura 4.1: Relação dos elementos de curvas f^{j+1} e \bar{f}^{j+1} .

4.1.1 Perturbações de um contorno

Se define como perturbações de um contorno f^j ao contorno de detalhe g^j que é a diferença entre f^{j+1} e f^j . Neste caso, as perturbações de um segmento f_k^j são dadas pela função g_k^j , definida por (4.8). A Figura 4.2(a) mostra a sobreposição de contornos f^j , mais lisa, e f^{j+1} , com perturbações em relação a f^j . A Figura 4.2(b) mostra a função de perturbações em todo o contorno de f^j , mapeada em todo o domínio (2^{j+1}) linear, representando o contorno f^j como uma linha reta.

Na realidade, o contorno f^j representa a média das perturbações em relação ao f^{j+1} , segundo a geração de multiresolução adotada.

As perturbações são geradas diretamente a partir dos coeficientes de detalhe reconstruídos, \bar{d}^{j+1} , segundo (4.3), para todo os segmentos do contorno. Como \bar{d}^{j+1} é um vetor, este será associado a um sistema de coordenadas local, tal como usados no processo de decomposição em multiresolução da curva, só que neste caso, é conveniente que o sistema de coordenada local de cada \bar{d}_r^{j+1} seja formado pelas retas tangente e normal do curva \bar{f}^{j+1} no ponto inicial do segmento r , já que um ponto de controle, \bar{c}_r^{j+1} , corresponde a um extremo inicial do segmento \bar{f}_r^{j+1} , e que c_r^{j+1} corresponde ao ponto inicial de f_r^{j+1} , segundo (4.2), (vide Figura 4.3). O lema a seguir formaliza a correspondência entre os elementos de \bar{f}_r^{j+1} e f_r^{j+1} .

Lema 4.1 *Um segmento \bar{f}_r^{j+1} converge em f_r^{j+1} , e vice-versa.*

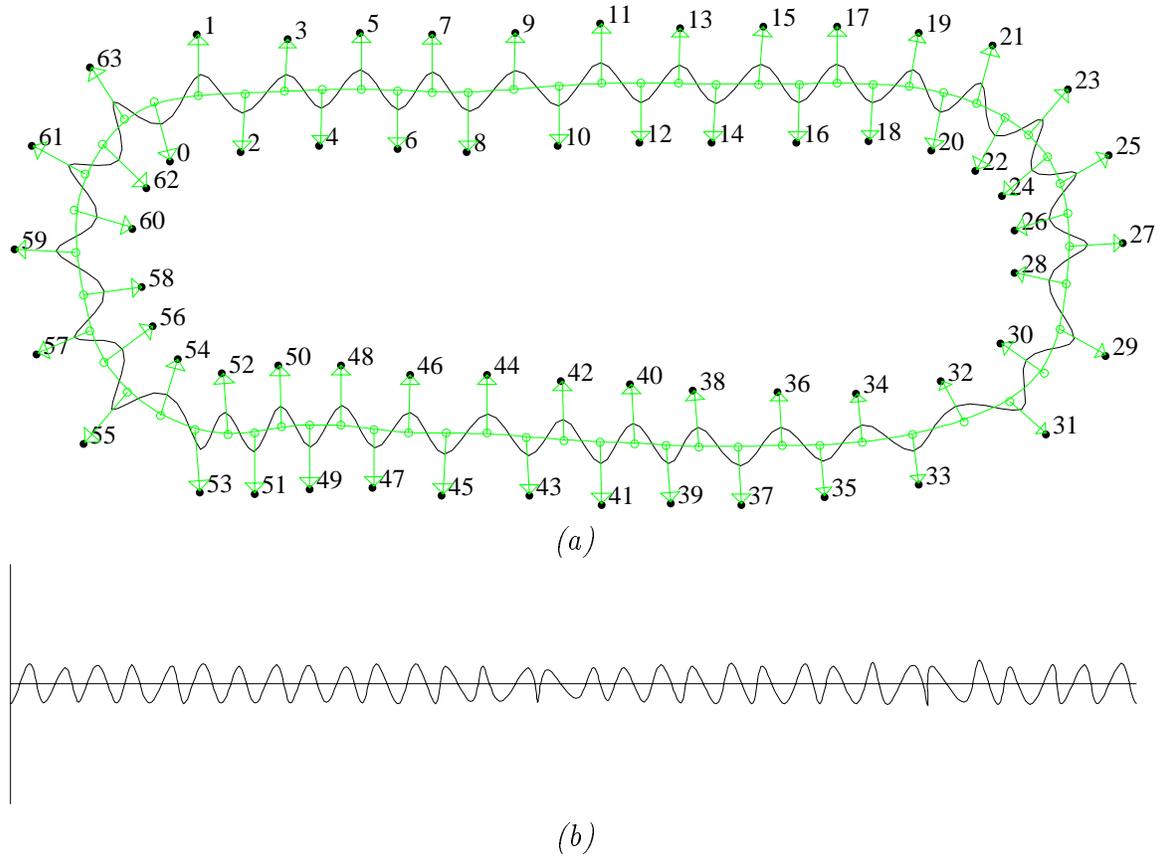


Figura 4.2: Perturbações de contorno de um objeto: (a) sobreposição de duas resoluções adjacentes de um contorno; (b) função de perturbação do contorno de menor resolução.

Prova: Seja um parâmetro real $0 \leq s \leq 1$, tal que define um outro segmento $\widehat{f}_{r,s}^{j+1}$, segundo (4.2), como

$$\widehat{f}_{r,s}^{j+1} = \tilde{N}(t) \left(\begin{pmatrix} \bar{c}_{r-1}^{j+1} \\ \bar{c}_r^{j+1} \\ \bar{c}_{r+1}^{j+1} \\ \bar{c}_{r+2}^{j+1} \end{pmatrix} + s \begin{pmatrix} \bar{d}_{r-1}^{j+1} \\ \bar{d}_r^{j+1} \\ \bar{d}_{r+1}^{j+1} \\ \bar{d}_{r+2}^{j+1} \end{pmatrix} \right), \text{ para } 0 \leq s \leq 1.$$

Como $0 \leq s \leq 1$, o limite pela esquerda da função é bem definido, sendo

$$\lim_{s \rightarrow -1} \widehat{f}_{r,s}^{j+1}(t) = f_r^{j+1}(t).$$

Assim, variando s de 0 para 1, obtem-se uma família de contornos que vá de $\bar{f}_r^{j+1} = \widehat{f}_{r,0}^{j+1}(t)$ para $f_r^{j+1} = \widehat{f}_{r,1}^{j+1}(t)$. Por tanto, $\bar{f}_r^{j+1}(t)$ converge em $f_r^{j+1}(t)$. Mesmo

critério é usado para a convergência de $f_r^{j+1}(t)$ em $\bar{f}_r^{j+1}(t)$, variando s de 1 para 0.

◇

Para formalizar a correspondência linear entre os elementos de segmentos de níveis adjacentes, se apresenta o teorema a seguir.

Teorema 4.1 *Um segmento f_r^j converge em $f_{2r}^{j+1} \cup f_{2r+1}^{j+1}$.*

Prova: A prova do teorema é imediata usando-se o lema anterior, já que um segmento é, segundo (4.5), $f_r^j = \bar{f}_{2r}^{j+1} \cup \bar{f}_{2r+1}^{j+1}$. Pelo Lema 4.1, o segmento \bar{f}_{2r}^{j+1} converge em f_{2r}^{j+1} e \bar{f}_{2r+1}^{j+1} converge em f_{2r+1}^{j+1} . Portanto, um segmento f_r^j converge em $f_{2r}^{j+1} \cup f_{2r+1}^{j+1}$.

◇

Como $c_r^{j+1} = \bar{c}_r^{j+1} + \bar{d}_r^{j+1}$ é um ponto no espaço, então \bar{d}_r^{j+1} é um vetor que tem sentido representativo no espaço, como mostrados na Figura 4.2(a) e Figura 4.3.

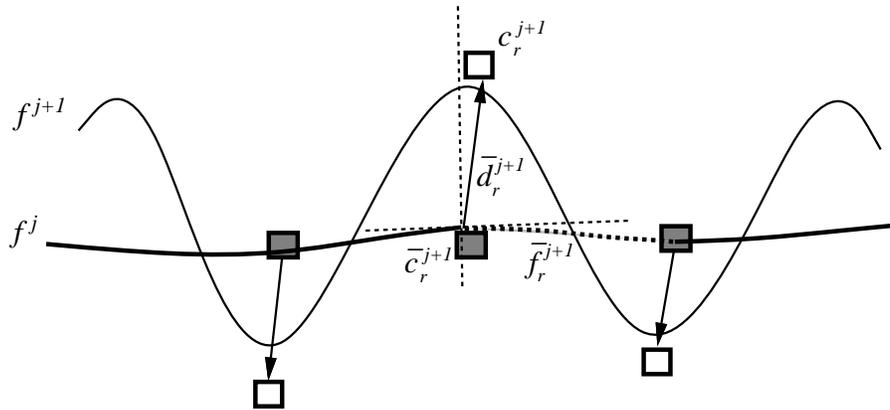


Figura 4.3: Definição de um sistema de coordenadas para \bar{d}_r^{j+1} .

4.2 Incorporação de perturbações nas caixas envolventes

As caixas hierárquicas em cada nível devem envolver os segmentos de contorno do objeto desse nível e suas perturbações, já que o objetivo é analisar as interferências entre fronteiras de objetos em forma graduada, controlando seus níveis de detalhe.

Com o enfoque seguido na construção de caixas envolventes do capítulo anterior, somente se logra cobrir por caixas hierárquicas os pedaços do contorno refinado, f^n . Em muitos casos, os pedaços de níveis inferiores, f^j , não são cobertos pelas caixas respectivas.

Uma representação de caixas envolvendo segmentos de cada nível de resolução e suas perturbações relativas permitirão uma análise efectiva de interferência por níveis dos objetos, já que se preserva em cada caixa as perturbações respectivas.

Uma caixa envolvendo um segmento de nível de resolução, f_r^j , não somente envolverá as perturbações deste nível, mas necessariamente deve envolver todas as perturbações de níveis mais refinados que convergem neste segmento. Como um segmento f_r^j pode ter perturbações g_r^j próximas a zero, mas as perturbações dessas perturbações, g_s^{j+1} , podem ser mais pronunciadas que as da anterior. Neste sentido, é conveniente incluir em cada caixa envolvendo f_r^j as perturbações de níveis mais refinadas convergentes neste segmento.

O método para incluir um segmento f_r^j e as perturbações refinadas em uma caixa consiste em, dada uma caixa, aumentá-la minimamente para os lados de forma que envolva o segmento f_r^j e suas perturbações refinadas. Aumentar uma caixa não implica modificar seus eixos; eles ficam constantes, o que varia são os lados e o centro da caixa.

Um critério trivial para aumentar a caixa envolvente é: dados os eixos de uma caixa, projetar todas as perturbações relativas ao segmento f_r^j sobre os eixos; depois, recalcular as dimensões da caixa com as projeções que escapam da caixa.

O critério trivial para aumentar as caixas é trabalhoso, porque por cada caixa, em cada nível, ter-se-ia que calcular as projeções das perturbações e segmentos. Felizmente, na transformação de wavelet prevalecem as médias. Assim, a representação do contorno no nível j é a média de seu equivalente refinado em $j + 1$. Este principio

vai facilitar o trabalho de aumentar uma caixa envolvente.

Teorema 4.2 *Uma caixa envolvendo um segmento de f^j e segmentos relativos de f^n também envolve as perturbações superiores a j .*

Prova: Analisando os elementos de multiresolução do objeto, o contorno f^j é uma média de suas perturbações. O contorno f^{j+1} oscila em torno de f^j dependendo das perturbações g^j . Nos níveis mais refinados, f^{n-1} é a média de g^{n-1} ; assim, f^n oscila em torno de f^{n-1} . As caixas envolvendo segmentos de f^{n-1} envolvem f^n . As caixas envolvendo f^{n-2} devem envolver f^{n-1} e f^n , isto implica que caixas envolvendo f^{n-2} e f^n envolvem também f^{n-1} . Este critério é aplicado, em geral, para qualquer resolução; as caixas envolvendo f^j e f^n envolvem as perturbações de níveis superiores a j .

◇

A Figura 4.4 ilustra um exemplo de uma caixa incluindo um segmento de f^{n-3} com suas perturbações refinadas.

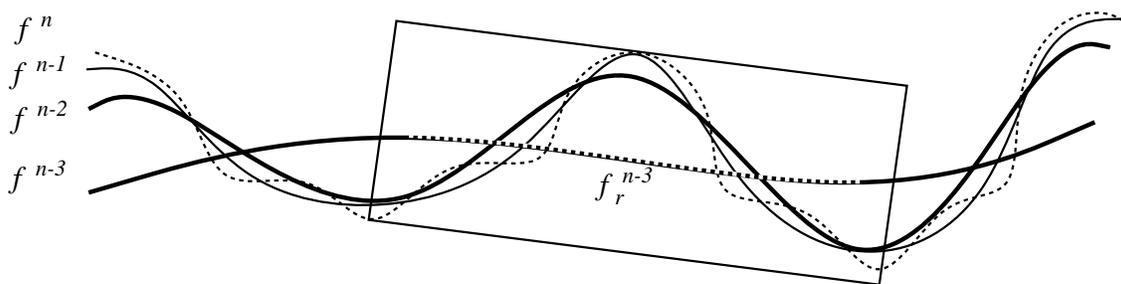


Figura 4.4: Caixa aumentada envolvendo segmento f_r^{n-3} e suas perturbações refinadas relativas.

Neste trabalho implementaram-se duas abordagens para aumentar caixas. Uma primeira é *caixa de caixas básicas*, que consiste em se calcular a caixa inicial como formulado no capítulo anterior, depois incluir o segmento de resolução relativo à caixa. Os resultados desta primeira abordagem são mostrados na primeira coluna (a) da Figura 4.6. Uma segunda abordagem é criar as caixas envolvendo segmentos em multiresolução depois aumenta-as. Determinar uma caixa envolvente como sendo a caixa básica do segmento f_r^j , aproveitando a resolução do segmento. Esta caixa é aumentada de forma que as caixas básicas relativas a este segmento estejam contidas na caixa recalculada. Os resultados deste segundo enfoque são apresentados na segunda coluna (b) da Figura 4.6.

Dos dois métodos utilizados para criar caixas envolvendo segmentos de superfície f_r^j e suas perturbações mais refinadas, o mais natural é o segundo; que determina a orientação da caixa envolvente pelo comportamento do segmento f_r^j . Segundo a Figura 4.6, a segunda coluna (b) mostra que as caixas são mais adaptadas aos segmentos correspondentes em cada resolução que as calculadas pelo primeiro método. Outra característica que se observa, nestas caixas envolventes, é que, na maioria das vezes, as caixas respectivas são menores do que as caixas similares construídas pelo outro método. De fato, ao se computarem as áreas totais das caixas em cada nível de resolução, verifica-se que, na maior parte dos casos, as áreas resultaram menores do que as do primeiro método.

Preservação de rugosidade do contorno refinado

O contorno mais refinado, f^n , ainda tem perturbações superiores. Essas perturbações, como qualquer perturbação mais refinada, podem ser incluídas nas caixas básicas. Mas existe um limite de operações e transformações de elementos do contorno mas refinados, assim como existe limite para decomposição em versões inferiores. As perturbações mais refinadas de f^n definem as *rugosidades* da superfície refinada. Estas rugosidades são aproximadas via parâmetros estatísticos, para evitar o uso excessivo de recursos computacionais, principalmente o espaço de armazenamento de coeficientes mais refinado demandam.

Como introduzida na Seção 2.4 e tratadas em detalhe no Capítulo 5, as perturbações do contorno refinado é equivalente a uma distribuição de vetores aleatórios

independentes $N(0, \sigma^2)$. Como o contorno não necessariamente é homogêneo, se considera a perturbação de cada segmento aproximadamente homogênea. Assim, para cada segmento de nível refinado existe uma distribuição de variância σ^2 .

A variância de uma distribuição normal está associada a um intervalo de confiança, estatisticamente falando. O intervalo de confiança, nomeado neste trabalho como *tolerância*, indica que um número maior de perturbações representativas estejam dentro dele. As tolerâncias de nível refinado serão associadas a cada caixa básica; por tanto, elas devem aumentar a caixa básica em seu valor respectivo.

Pode-se imaginar cada segmento básico como tendo espessura igual a duas vezes o valor da tolerância; portanto, este segmento espesso deve estar completamente incluído na caixa básica. Para tal, se adicionará o tamanho de tolerância correspondente ao segmento básico nos pontos tangentes às arestas da caixa básica.

Logicamente, não é possível idealizar-se o contorno refinado com blocos de rugosidades constantes. Mas para efeitos de determinação de aproximações ou contatos entre os objetos, a tolerância concebida desta forma faz um papel importante. Já na etapa de tratamento de contatos, as tolerâncias definem uma distribuição não homogênea de rugosidade no contorno de cada intervalo, dependendo das tolerâncias dos segmentos vizinhos (detalhes no Capítulo 5).

As tolerâncias de caixas superiores podem ser herdadas a partir das tolerâncias de suas respectivas folhas básicas podendo adotar-se a média, a menor, ou a maior das tolerâncias das caixas básicas que estão sendo incluídas nesta caixa superior. Essas tolerâncias vão depender do nível de tratamento de contatos, já que estes podem demandar como sendo contatos entre objetos de forma superficial ou aproximada.

4.3 Avaliação de construção de MOBBtree

A construção de todos os MOBBtrees, uma para cada objeto da cena, é feita em uma etapa de pre-processamento; ou seja, antes de iniciar com a iteração de animação. Antes de construir a árvore relativa a um objeto, deverão ser geradas as representações em multiresoluções do objeto via a transformada de wavelet. Com a ajuda dessas resoluções, deve-se construir a respectiva MOBBtree. Assim, a estrutura de dados descrevendo um objeto inclui um apontador para a estrutura do MOBBtree relativo

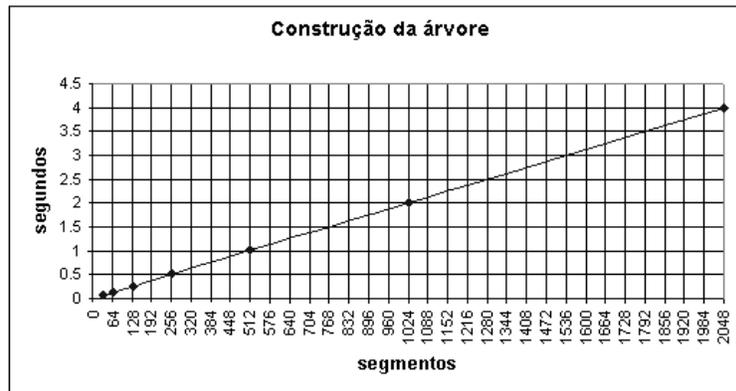
ao objeto.

Para calcular o número c de caixas de uma árvore que vai demandar um objeto de $n = 2^k$ segmentos, se parte do fato que o processo *bottom-up* de criação da árvore executa $n/2^j$ iterações para construir todas as caixas do nível j . Assim, o nível da folha terá n iterações, o segundo nível terá $n/2$ iterações, etc. Considerando que cada caixa é construída em tempo constante, a construção de uma árvore é feita em tempo linear, $O(n)$, já que o número de caixas de uma árvore é dado por

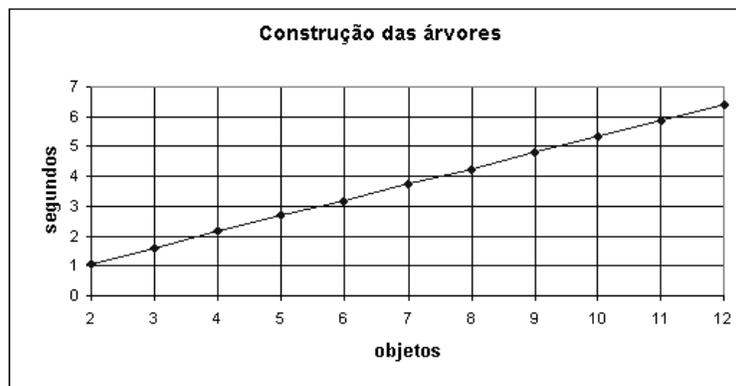
$$c = n + \frac{n}{2} + \frac{n}{4} + \dots + \frac{n}{n}$$

$$c = 2n - 1.$$

Os quadros mostrados na Figura 4.5 indicam o comportamento das construções das árvores com respeito ao tempo, em segundos. Os processos de medição de tempo foram feitos na estação *Sun SPARCstation 20* com *Solaris*, sendo o tempo tomado o mínimo dos tempos de várias repetições da cada caso para minimizar o tempo adicional devido à execução de outros processos. A Figura 4.5(a) mostra o comportamento linear na construção de árvores de objetos de múltiplos de 32 segmentos. Assim, o primeiro ponto corresponde a um objeto de 32 segmentos, o segundo a um objeto de 64 segmentos, etc. A Figura 4.5(b) mostra o comportamento linear de construção de árvores variando o número de objetos de 256 segmentos cada um, onde o primeiro ponto corresponde a construção das árvores de dois objetos, o segundo de três objetos, e o último a doze objetos. Com essas tendências pode-se ter uma idéia de quanto demoraria a construção de uma árvore, por exemplo, de 4096 segmentos.

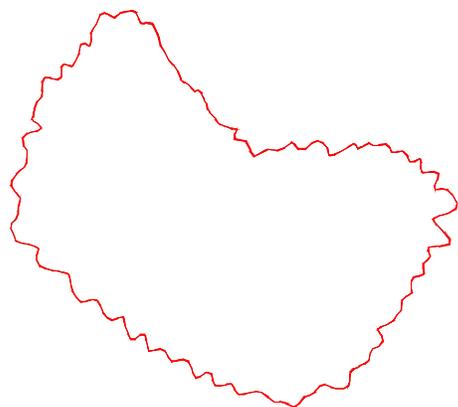


(a)

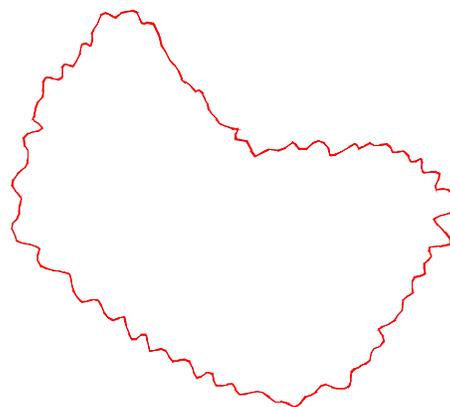


(b)

Figura 4.5: Construção de MOBBtree em segundos: (a) um objeto com número de segmentos variável; (b) objetos de 256 segmentos cada um.

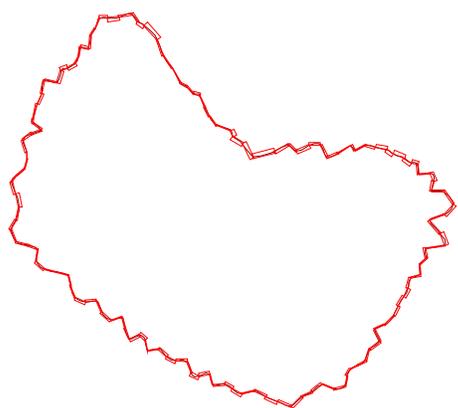


(a-1) caixas básicas

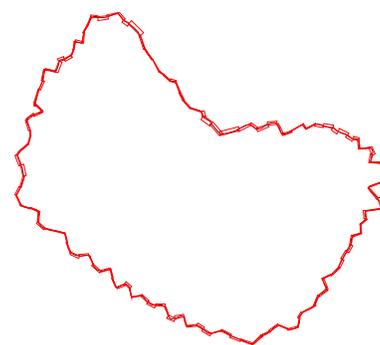


(b-1) caixas básicas

(nível 8)

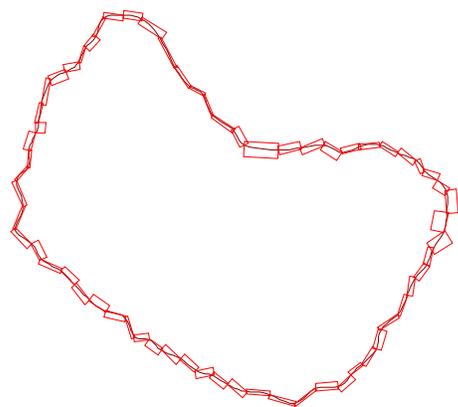


(a-2) área das caixas 898.001

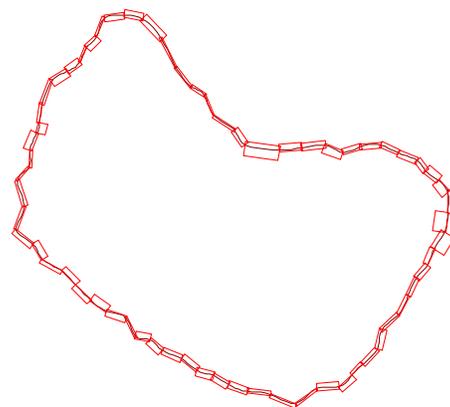


(b-2) área das caixas 877.592

(nível 7)

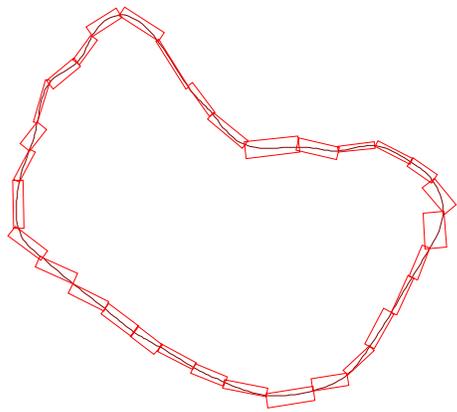


(a-3) área das caixas 2461.882

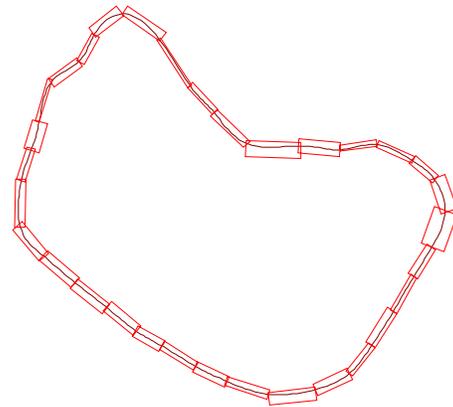


(b-3) área das caixas 2404.968

(nível 6)

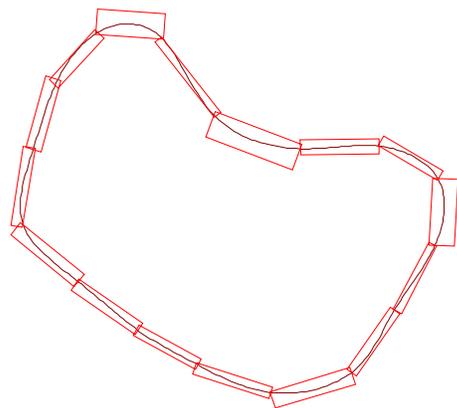


(a-4) área das caixas 3595.044

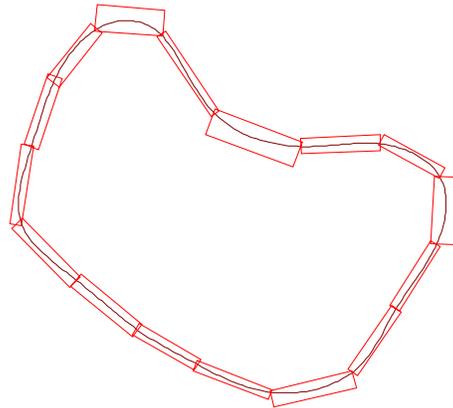


(b-4) área das caixas 3852.326

(nível 5)

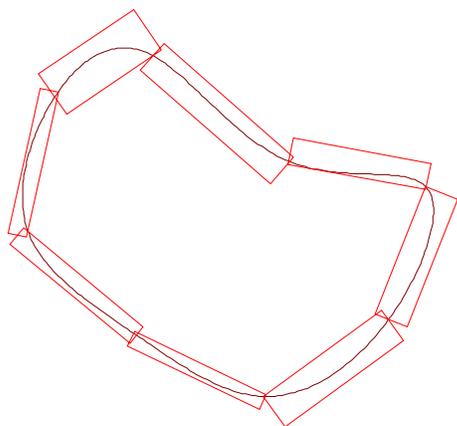


(a-5) área das caixas 5302.826

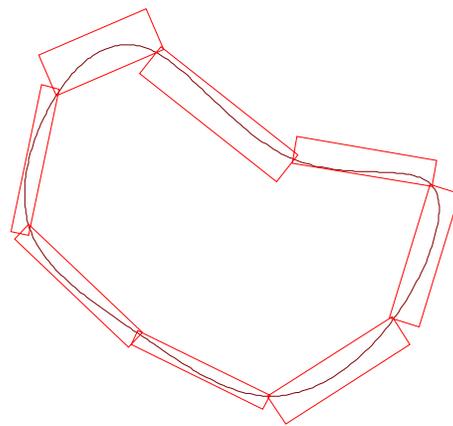


(b-5) área das caixas 5422.778

(nível 4)



(a-6) área das caixas 10161.296



(b-6) área das caixas 9466.018

(nível 3)

Figura 4.6: Geração de caixas envolventes de segmentos de curvas em multiresolução.

Capítulo 5

Relação entre as perturbações e as tolerâncias

Uma vez que duas caixas orientadas, B_a^i e B_b^j , relativas aos contornos em uma determinada resolução dos objetos, estejam se interpenetrando, determinam-se as três situações em que os objetos, neste segmento, podem estar: contato, interpenetrados ou separados. Essas três situações são determinadas por meio da verificação analítica de interferências dos respectivos contornos, considerando-se um pequeno intervalo entre os segmentos dos contornos. A Figura 5.1 mostra as possíveis situações que podem assumir dois segmentos de contornos com seus respectivos intervalos, chamados intervalos de tolerância.

Os objetos podem possuir contornos irregulares, por isto cada segmento pode ter uma média de aspereza diferente das demais. Neste sentido, é necessário associar o grau de perturbações de cada segmento a um intervalo de tolerância, que deve ser incluído nas respectivas caixas envolventes.

Um objeto de geometria arbitrária considerado em uma determinada resolução, como explicado no capítulo anterior, possui perturbações constituídas pelas versões superiores, as quais devem ser levadas em conta para definir as tolerâncias na resolução respectiva. Neste capítulo se definem as tolerâncias associadas a cada segmento, em cada resolução em que pode ser expresso um objeto, que são associadas às perturbações de seus respectivos contornos, o que será útil na análise de contatos dos objetos em animação.

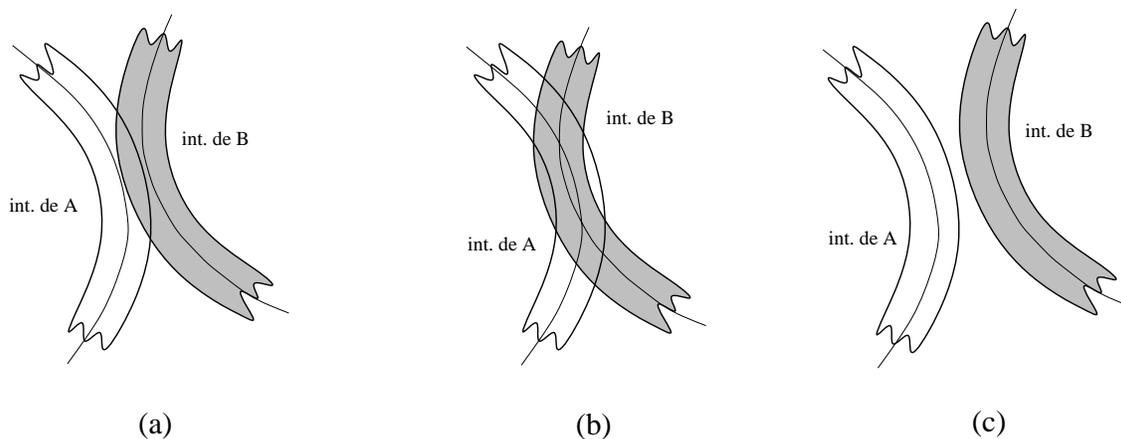


Figura 5.1: *Situações entre dois objetos: (a) contato entre A e B; (c) A e B interpenetrados; (b) A e B separados.*

5.1 Intervalos de tolerância

Se o contorno f , expressa em uma determinada resolução, tivesse uma perturbação uniforme em toda sua extensão, estabelecer-se-ia uma tolerância constante em torno dela, para sua comparação de proximidades com outros objetos, já que, na prática, não existe um contato tangencial exato entre dois contornos. Deste modo, considerar uma tolerância em torno de cada contorno é importante na verificação de interferências e, particularmente, na detecção de contatos.

No mundo real, nem todos os objetos têm superfícies absolutamente lisas; sempre possuem perturbações não homogêneas que definem as rugosidades em suas superfícies. A Figura 5.2, tomada de [HallResn67], mostra um exemplo de como um contorno, considerado finamente polido, possui irregularidades. Então, considerar que a superfície dos objetos, em animações envolvendo dinâmica, é perfeitamente lisa, constitui uma limitação que pode afetar seus comportamentos.

Um contorno com perturbações irregulares pode ser localmente aproximada para um contorno com certo grau de rugosidade ocorrendo dentro de um intervalo de tolerância. Assim, é possível encontrar-se, em alguns segmentos do contorno, valores pequenos de tolerância, e, em outros segmentos, apresentarem-se maiores magnitudes de tolerância.

O intervalo de tolerância, considerado neste trabalho, vai manter uma simetria

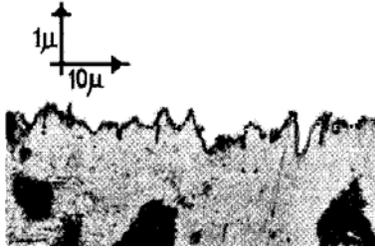


Figura 5.2: Vista ampliada de uma seção de aço finamente polida. Esta imagem aparece no texto *Physics, Holliday & Resnick, 1967*.

entre o limite exterior e interior da tolerância com relação ao contorno f . Esta suposição ocorre porque f é a média de um outro contorno refinado f^{+1} , relação definida pela transformada wavelet discutida no Capítulo 2. O contorno f^{+1} é a média de f^{+2} , caso houver, e é a perturbação de f^{+1} e de f . Em geral, denota-se por f o contorno de referência em um nível de resolução arbitrário que corresponde à média de suas perturbações f^+ .

Caso não existam perturbações sobre um segmento do contorno, considera-se um intervalo mínimo de tolerância (Figura 5.3(a)). Caso exista, a tolerância é definida pelas respectivas perturbações. As perturbações, sobre um pedaço de contorno, podem ter comportamentos constantes ou variados (Figura 5.3(b) e (c) respectivamente), que definem tolerâncias fixas ou variadas, respectivamente.

5.1.1 Enfoque estatístico de intervalo de tolerância

Determinar o intervalo de tolerância em torno de f a partir de suas perturbações não é nada simples. Os estimadores estatísticos servem como uma ferramenta para determinar esse intervalo a partir das perturbações, de modo a criar um espaço de tolerância em torno de f , onde exista uma maior ocorrência de perturbações. Podem existir algumas poucas perturbações saindo dessa tolerância, de modo que elas não serão significativas. Em vista da simetria das perturbações em relação a f , para efeitos de cálculo, o intervalo de tolerância é representado por uma tolerância $\lambda > 0$

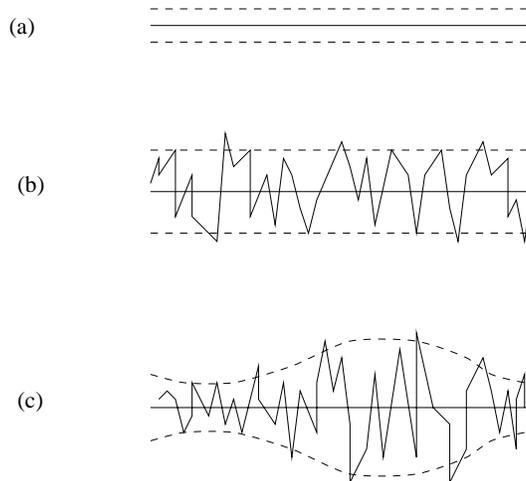


Figura 5.3: *Casos possíveis para o engrossamento de um segmento de contorno: (a) sem perturbação; (b) perturbação quase constante; (c) perturbação variada.*

para a distância em relação a f .

Do ponto de vista estatístico, o intervalo de tolerância, conceituado por Donoho e Johnstone [DonoJohn94] e Abramovich e Benjamini [AbraBenj96] para a estimação de curvas não paramétricas a partir de outra contaminada por ruídos, é relacionado ao teste de hipótese, onde se valida se cada elemento da amostra está suficientemente próximo a um valor pequeno (quase zero) ou não. Na estimação de curvas, o princípio é a poda dos elementos que caem dentro do intervalo de tolerância, com os segmentos externos se definem os atributos da curva. Critério similar acontece na compressão de imagens com wavelets [StDeSa96, FerPer96, Sweldens96].

Na abordagem de Donoho e Johnstone, os n elementos da amostra são os coeficientes de wavelet da curva com ruídos que são usados para se computar a variância σ que se relaciona com o valor da tolerância λ pela formulação da *tolerância universal*, proposta por eles, como

$$\lambda_{un} = \frac{\sqrt{2 \log n}}{\sqrt{n}} \sigma, \quad (5.1)$$

A formulação de tolerância universal é parecida com a expressão de “intervalo de confiança” [JohnWich92] que tem a forma

$$\lambda = q\sigma, \quad (5.2)$$

para $0.0 \leq q \leq 1.0$. Quando q se aproxima de 1.0, uma maior porção dos n valores

amostrados caem dentro do intervalo definido. Um comportamento análogo acontece com $\frac{\sqrt{2 \log n}}{\sqrt{n}}$ quando n é pequeno, não para valores grandes de n . Neste trabalho, obteve-se por considerar (5.2), com $q = 0.95$, para relacionar λ com σ , para qualquer n .

5.2 Tolerâncias de segmentos de contorno refinado

Chama-se de contorno refinado ao contorno f de máxima resolução expressa geometricamente. As supostas resoluções superiores f^+ podem ser aproximadas proceduralmente, inserindo pontos de controle, como visto no Capítulo 2, mas isso implica em maiores espaços de memória para seu armazenamento. Outra abordagem, adotada neste trabalho, é considerar que essas supostas resoluções superiores, definindo perturbações do contorno refinado, ocorrem com maior probabilidade dentro de um intervalo de tolerância, simulando rugosidades homogêneas.

Para se aproximar as perturbações irregulares do contorno refinado, considera-se que as tolerâncias ocorrem para cada segmento do contorno. Supõe-se que, sobre o segmento, ocorrem rugosidades homogêneas, mas que podem sofrer uma pequena variação influenciada pelos comportamentos das rugosidades dos segmentos adjacentes. Com este critério, se associa um valor de intervalo de tolerância λ_j a cada segmento f_j . Assim, o contorno refinado dos objetos terá uma variedade de tolerâncias.

A forma de distribuição de tolerâncias sobre f pode ser dada aleatoriamente ou por uma função definida a partir de uma biblioteca de tolerâncias estabelecida pelo usuário. A biblioteca de tolerâncias, criada para testes neste trabalho, considera m valores λ_i (por conveniência toma-se $m = 4$) tal como apresentados na Figura 5.4, que devem ser atribuídos a cada segmento básico do contorno refinado do objeto. Um exemplo da função de distribuição estabelecida pode ser: distribua λ_i da biblioteca de tolerâncias *tipo_r* em cada k segmentos adjacentes (k pode ser gerado por uma função definida ou pode ser uma constante externa). A Figura 5.5 mostra, como exemplo, o quadro de distribuição de tolerância *tipo_3* e *tipo_5* da biblioteca nos contornos de 32 e 128 segmentos, respectivamente. Na Figura 5.7(b), o objeto esquerdo de geometria arbitrária, tem perturbações geradas com tolerâncias *tipo_3* e a esfera, lado direito, tem perturbações geradas com tolerâncias *tipo_5*. A Figura 5.6 ilustra graficamente um tipo de distribuição de tolerâncias de *tipo_3* da Figura 5.5, onde cada barra é

relacionada com o valor da tolerância atribuída a um segmento.

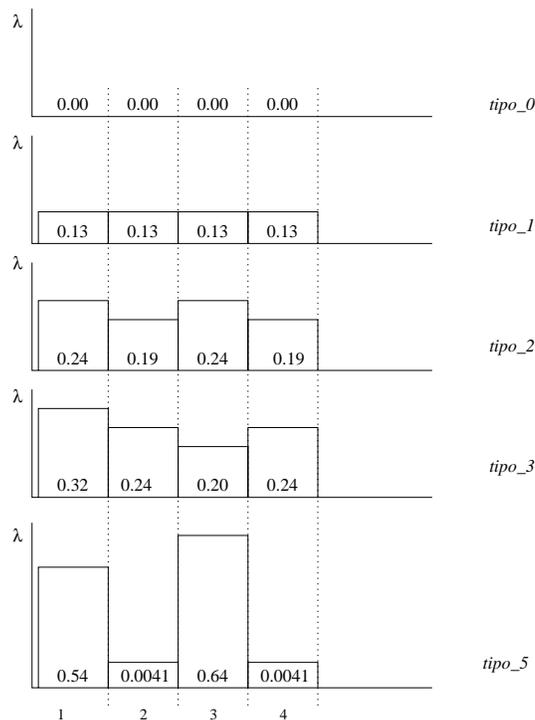


Figura 5.4: Exemplo de biblioteca de quatro tipos de tolerâncias.

Geração de perturbações no contorno

Para visualização de contornos refinado com rugosidades é necessário gerar aleatoriamente as perturbações baseada nas tolerâncias, e distribuí-las sobre cada segmento do contorno. Os n valores que definem as perturbações são os vetores perpendiculares a certos pontos, uniformemente distribuídos, sobre o contorno refinado do objeto. A normal da tangente no ponto dado define a direção do vetor e a magnitude é fornecida pelo gerador de valores aleatórios $N(0, \sigma^2)$. O parâmetro σ é obtido da expressão de intervalos de confiança (5.2), já que é conhecido o valor de λ para cada segmento. Este processo não afeta o comportamento dos objetos em animação, já que explicitar as rugosidades do contorno dos objetos afeta somente a visualização.

A parte (a) da Figura 5.7 mostra os objetos com contornos refinados sem perturbações, (b) mostra os objetos com as perturbações geradas usando-se a biblioteca de tolerâncias e a distribuição probabilística aleatoria $N(0, \sigma^2)$ com n perturbações em

<i>segmentos</i>	<i>tipo_3</i>	<i>tipo_5</i>
0	0.32	0.55
1	0.24	0.55
2	0.20	0.55
3	0.24	0.55
4	0.32	0.04
5	0.24	0.04
6	0.20	0.04
7	0.24	0.04
8	0.32	0.65
9	0.24	0.65
10	0.20	0.65
11	0.24	0.65
12	0.32	0.04
13	0.24	0.04
14	0.20	0.04
15	0.24	0.04
16	0.32	0.55
17	0.24	0.55
18	0.20	0.55
19	0.24	0.55
20	0.32	0.04
21	0.24	0.04
22	0.20	0.04
23	0.24	0.04
24	0.32	0.65
25	0.24	0.65
26	0.20	0.65
27	0.24	0.65
28	0.32	0.04
29	0.24	0.04
30	0.20	0.04
31	0.24	0.04
-	—
127	—	0.04

Figura 5.5: Exemplo de um tipo de distribuição de tolerâncias da biblioteca em um contorno de 32 e 128 segmentos, respectivamente.

cada segmento do contorno. As n perturbações distribuídas sobre um segmento f_j são computadas pelo gerador de valores aleatórios Gaussianos, [PrTeVe+92].

5.3 Tolerância em multiresolução

A análise de contatos pode ser feita em uma resolução superior que não seja a refinada do objeto. Nesta situação, as tolerâncias devem ser computadas diretamente a partir das perturbações do contorno deste nível, porque tem-se a informação da descrição geométrica do objeto.

Pode-se considerar como contorno em análise o objeto limitado por uma versão menos refinado do contorno, como f^{n-1} ; neste caso, as perturbações definindo a tolerância de cada segmento desse contorno são definidas pelos elementos de detalhe correspondentes a essa resolução. A Figura 5.8 mostra a versão refinada de um objeto irregular de 128 segmentos, suas representações na resolução imediata inferior e o detalhe que define sua perturbação.

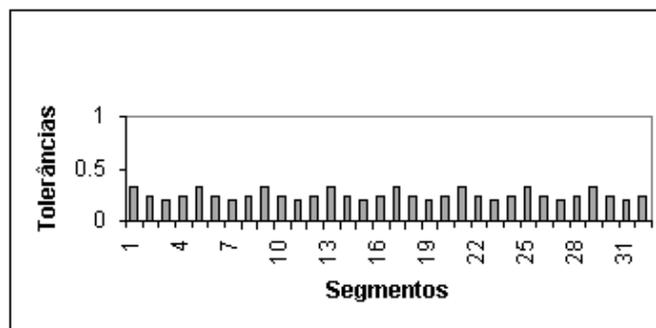


Figura 5.6: *Exemplo de distribuição de tolerâncias tipo_3, com uma função de distribuição, sobre um contorno de 32 segmentos.*



Figura 5.7: *Dois objetos definidos: (a) sem mostrar detalhe de aspereza de seus contornos; (b) mostrando asperezas de seus contornos segundo a biblioteca de tolerâncias.*

Os intervalos de tolerâncias associados a cada segmento de um contorno em multiresolução podem ser computados diretamente dos coeficientes de wavelets ou a partir dos contornos de detalhe dessa resolução.

5.3.1 Tolerâncias definidas por coeficientes wavelets

Como os coeficientes de detalhe D^{n-1} definem a função de detalhe g^{n-1} , de modo que $f^n = f^{n-1} + g^{n-1}$, como visto no Capítulo 4, pode-se aproximar a tolerância relativa a cada segmento do contorno f^{n-1} com os coeficientes de wavelets respectivos, uma

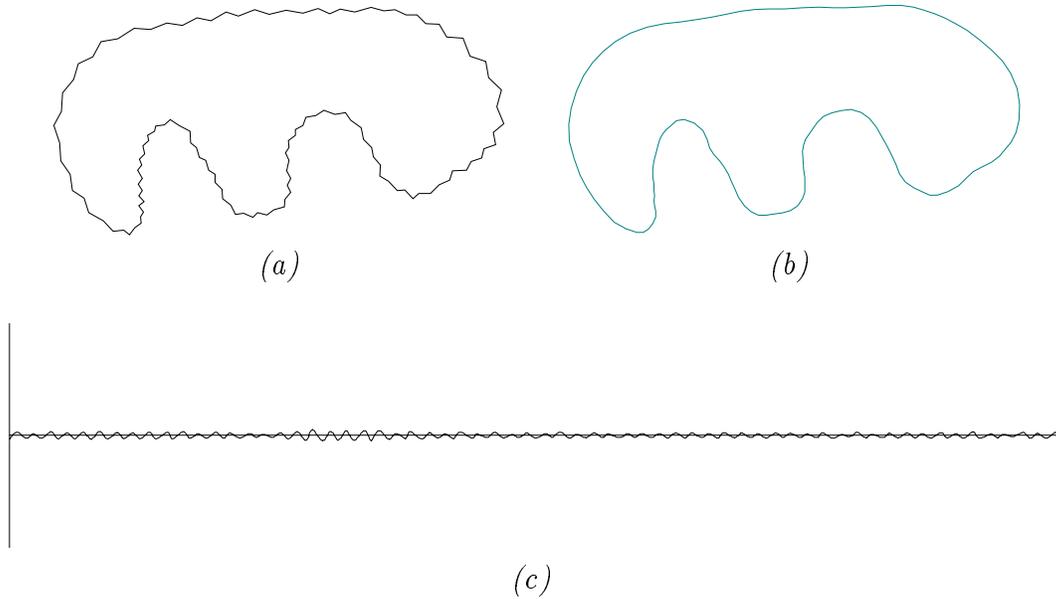
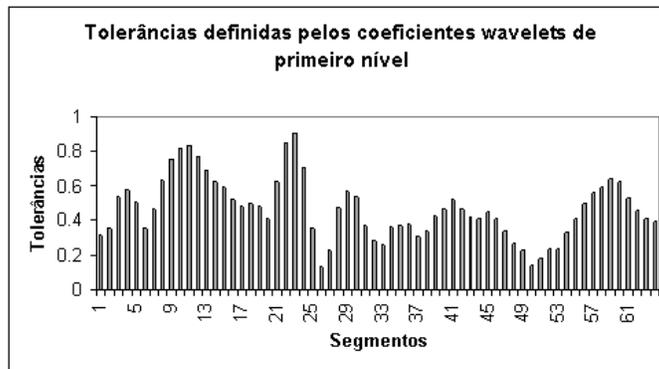


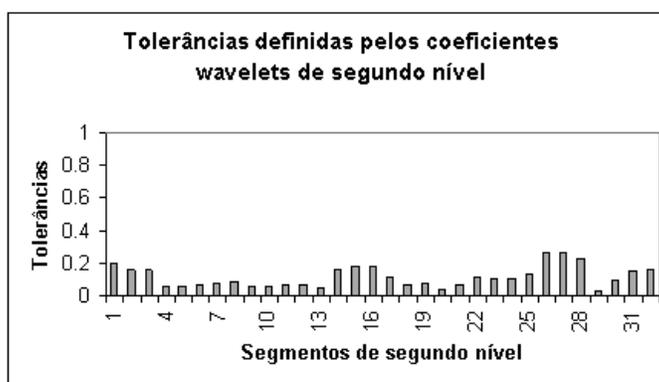
Figura 5.8: Objeto mais refinado define perturbações de sua representação menos refinada: (a) versão refinada; (b) versão menos refinada; (c) contorno de detalhe mapeado horizontalmente.

vez que se tem os coeficientes de wavelets explicitados neste nível de resolução.

Segundo o teorema 4.1, um segmento f_j^{n-1} é diretamente associado a três pontos de controle para se projetar a um nível imediato superior, f^n . Esses pontos de controle são: dois pontos extremos do segmento, e um ponto que se insere dentro do segmento. Por isto, pode-se considerar como mínimo três coeficientes wavelets relativos ao segmento em análise, que são os vetores: d_{j-1}^{n-1} , d_j^{n-1} e d_{j+1}^{n-1} . Já que estes estão orientados localmente com respeito à tangente no ponto médio de seus respectivos segmentos (como descrito no Capítulo 2), pode-se estimar a variância σ , com respeito a média considerada como zero, com a componente normal de cada vetor, para depois computar a tolerância λ_j^{n-1} aplicando (5.2). Podem-se considerar mais de três coeficientes de wavelets vizinhos para se obter um resultado mais coerente com a geometria local do segmento. No caso geral, pode-se considerar k elementos para ambos os lados de d_j^{n-1} , sendo no total $2k + 1$ elementos. A Figura 5.9(a) mostra, através das barras, as tolerâncias de 64 segmentos da primeira versão inferior do objeto apresentado na Figura 5.8.



(a)



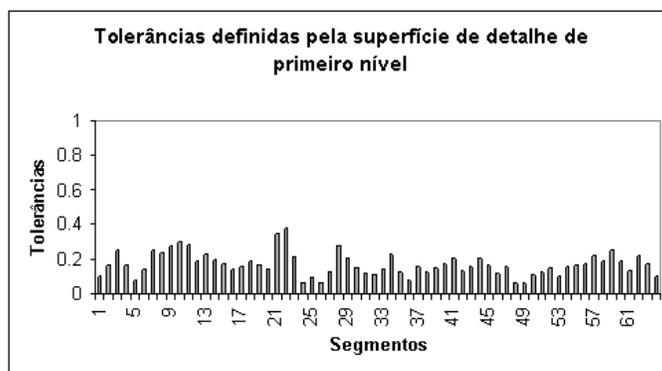
(b)

Figura 5.9: Seqüência de tolerâncias definidas pelos coeficientes wavelets do objeto da Figura 5.8: (a) primeiro nível grosseiro; (b) segundo nível grosseiro.

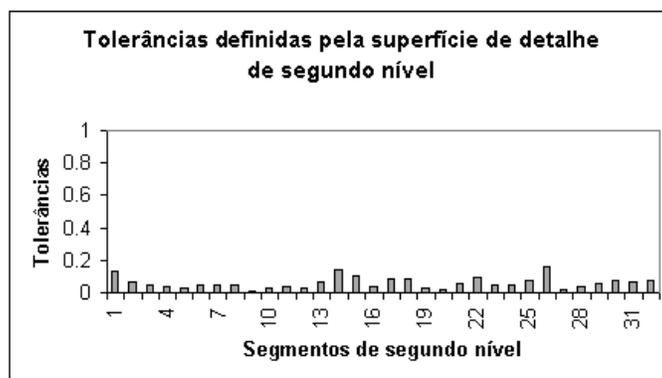
5.3.2 Tolerâncias definidas pelo contorno de detalhe

Uma outra alternativa é computar λ_j^{n-1} a partir do contorno de detalhe g^{n-1} , conforme explicado na Seção 4.1, especificamente em (4.3), já que esta determina o contorno de perturbações de f^{n-1} . Sendo assim, pode-se determinar os k (como mínimo três) valores máximos e mínimos de g^{n-1} relativos a f_j^{n-1} para se estimar a variância σ . Este método permite o cálculo da tolerância relativa ao segmento f_j^{n-1} em forma mais exata do que a estimada usando-se os coeficientes de wavelets (D^{n-1}), porque inclui a distância de um contorno superior ao inferior, enquanto o método baseado em coeficientes wavelets computa tolerância aproximada, já que eles representam

as variações no nível de pontos de controle. A Figura 5.10 mostra as tolerâncias calculadas a partir dos contornos de detalhe, (a) para primeiro nível, e (b) para segundo nível, do objeto definido na Figura 5.8.



(a)



(b)

Figura 5.10: Seqüência de tolerâncias definidas pelo contorno de detalhe do objeto da Figura 5.8: (a) primeiro nível grosseiro; (b) segundo nível grosseiro.

Comparando os resultados, mostrados nas Figuras 5.9 e 5.10, observa-se que eles são similares e que o resultado no caso da tolerância baseada nos coeficientes wavelets são uma versão escalada (maior) daqueles em que a tolerância é definida pelos contornos de detalhe respectivas.

5.3.3 Tolerâncias atribuídas aos segmentos de resoluções inferiores

Para se definir a tolerância total, λ_j , atribuída ao segmento f_j^{n-1} , deverão ser consideradas as tolerâncias dos segmentos relativos ao nível superior. Como são dois segmentos adjacentes de f^n que convergem em f_j^{n-1} , então as tolerâncias relativas são λ_{2j}^n e λ_{2j+1}^n . É conveniente se considerar a média aritmética dessas tolerâncias, sendo esta dada por λ_j^M . Com isto, a tolerância associada ao segmento f_j^{n-1} é dada por

$$\lambda_j = \lambda_j^{n-1} + \lambda_j^M. \quad (5.3)$$

Para as resoluções mais inferiores, f^{n-r} com $r > 1$, pode-se aplicar critério similar aos explicados nas seções anteriores, usando a expressão (5.3) para o cálculo da respectiva tolerância com λ_j^{n-r} ao invés de λ_j^{n-1} . A Figura 5.11 mostra um esquema gráfico da definição de uma tolerância em um nível de resolução j , adicionados as tolerâncias de resoluções superiores.

5.3.4 Limitações de contornos de referências

Analisando as seqüências de transformações de um objeto em múltiplos níveis de resolução, observam-se as transformações dos objetos originais para seus equivalentes em versões grosseiras (visto no Capítulo 2), cada vez com maiores diferenças. Um objeto expresso em resoluções superiores mantêm as características originais. Isto se explica porque os coeficientes wavelets em resoluções menores são mais variados em tamanho, o que faz com que possam desvirtuar exageradamente os valores de tolerâncias para essas resoluções.

Por exemplo, a Figura 2.7 mostra que as características de forma de um objeto são progressivamente perdidas à medida que as representações se tornam mais grosseiras.

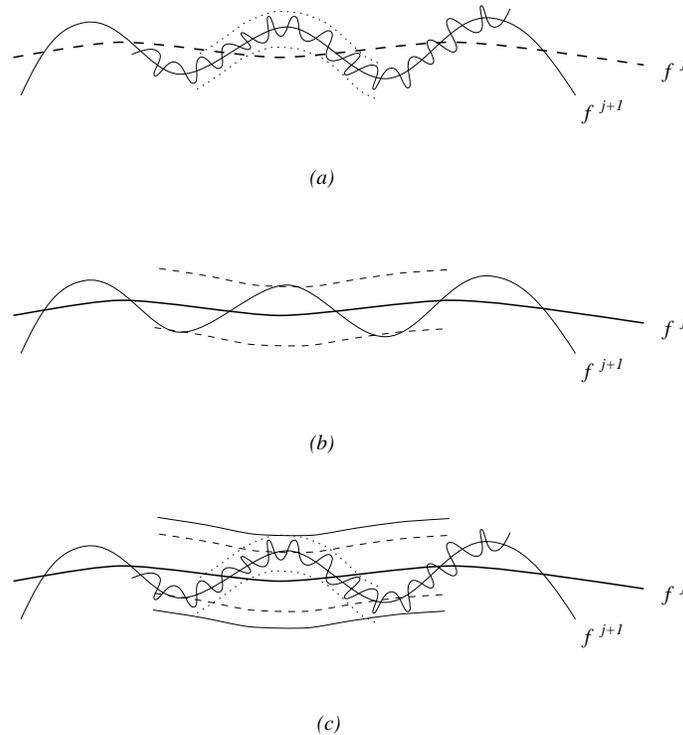


Figura 5.11: Definição da tolerância do nível j : (a) tolerância relativa à resolução $j + 1$; (b) tolerância definida pela perturbação de j ; (c) tolerância considerando todas as perturbações superiores.

5.4 Tolerância relativa a um ponto

Os intervalos de tolerância podem variar em toda o contorno do objeto, inclusive dentro de um segmento básico. Para facilitar a detecção de contatos, considerou-se a tolerância constante em cada segmento do contorno, mas para o tratamento dos mesmos, a tolerância varia em cada ponto do segmento.

Para cada ponto do contorno existe uma tolerância $\lambda(t)$, que é calculada por interpolações. Nesta etapa, o que interessa é o tamanho de $\lambda(t)$ relativo ao ponto $f(t_c)$. Por isso, a função $\lambda(t)$, que define a tolerância, deve ser contínua. Assim, para simular cada segmento da função de tolerância, deve ser usada alguma forma de interpolação, que pode ser uma simples interpolação linear, ou uma interpolação polinomial de grau maior que linear. A fim de se obter uma simulação da função de tolerância comportada, ao menos localmente, considerou-se, neste caso, a interpolação

polinomial cúbica [PrTeVe+92]. Embora implique na resolução de um sistema de quatro equações lineares a se resolver para cada segmento, o cálculo dos coeficientes do polinômio é facilitado pela característica da distribuição contínua dos pontos do domínio unitário da curva, como mostra a Figura 5.12(c).

Um segmento de função polinomial cúbica de interpolação é definido por

$$\gamma(t) = a_0 + a_1t + a_2t^2 + a_3t^3. \quad (5.4)$$

Como o domínio de cada segmento f_j é de amplitude constante, e o valor λ_j é associado ao ponto médio do segmento, o domínio de cada função $\gamma_k(t_j)$ é trasladado do domínio de f_j . Em particular, o domínio de f_j é um intervalo $[0, 1)$, enquanto o domínio de um subsegmento significativo de γ_k é $[t_{k-1}, t_k)$ com $t_k - t_{k-1} = 1$. Os subsegmentos extremos γ_k não são significativos, porque não recebem a influência dos outros valores mais à direita ou mais à esquerda do conjunto de quatro valores considerados para a interpolação; as Figuras 5.12(a) e (b) mostram esses casos. Por isto, sem perder a generalidade, se define a função de tolerância para os pontos $t_0 = 0, t_1 = 1, t_2 = 2$ e $t_3 = 3$, de modo que

$$\begin{aligned} \gamma_k(t_0) &= \lambda_{k-1}, \\ \gamma_k(t_1) &= \lambda_k, \\ \gamma_k(t_2) &= \lambda_{k+1}, \\ \gamma_k(t_3) &= \lambda_{k+2}, \end{aligned}$$

onde k como t são computados da maneira formulada a seguir:

$$\begin{aligned} \text{se } (t_c < 0.5) \\ & k = j - 1 \\ & t_\lambda = t_c + 1.5 \\ \text{senao} \\ & k = j \\ & t_\lambda = t_c + 1.0. \end{aligned}$$

Estes cálculos são necessários porque $\gamma_k(t)$ é defasado em relação a $f_j(t_c)$, e cada λ_j é associado a $f_j(0.5)$. A partir desses dados se definem as equações lineares de interpolação polinomial cúbica, avaliando (5.4) nos quatro pontos vizinhos, $\{t_i\}_{i=0,\dots,3}$,

sendo sua formulação matricial

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \mathbf{M}^{-1} \cdot \begin{bmatrix} \lambda_{k-1} \\ \lambda_k \\ \lambda_{k+1} \\ \lambda_{k+2} \end{bmatrix}. \quad (5.5)$$

Assim, calcular os coeficientes de (5.4), para cada segmento, é simples porque a matriz quadrada \mathbf{M}^{-1} é constante e única para qualquer t_c do contorno do objeto, variando unicamente os valores do vetor de tolerâncias (λ_i). A matriz \mathbf{M}^{-1} é definida pelos valores de $\{t_i\}_{0,\dots,3}$ como

$$\mathbf{M}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \end{bmatrix}^{-1} = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ -1.8333 & 3.0 & -1.5 & 0.3333 \\ 1.0 & -2.5 & 2.0 & -0.5 \\ -0.1667 & 0.5 & -0.5 & 0.1667 \end{bmatrix}.$$

5.5 Atrito adotado do grau de intervalo de tolerância

No contato entre dois contornos sempre existe uma força de atrito (explica-se em detalhe no Capítulo 7), para o qual requer-se um coeficiente μ , que representa o coeficiente de atrito em função das propriedades físicas do contorno e de sua vizinhança.

O atrito, estudado num nível microscópico, é um fenômeno muito complicado e as leis utilizadas são de carácter empírico e fornecem predições aproximadas. O atrito depende de muitas variáveis ([HallResn67]), tais como a natureza dos materiais, irregularidades das superfícies, películas superficiais, temperatura e extensão da contaminação e desgaste da área de contato. Por tudo isso, não é surpreendente que não exista uma teoria exata para atrito seco e que as leis do atrito sejam empíricas.

Neste trabalho, simula-se o atrito associado a cada ponto de contato, a partir das irregularidades dos contornos, sem pretensão de descobrir a essência mesma do atrito, mas apenas aproximar o comportamento de uma dinâmica de contato entre contornos de materiais idealizados, sem desgastes, com resultados visualmente aceitáveis.

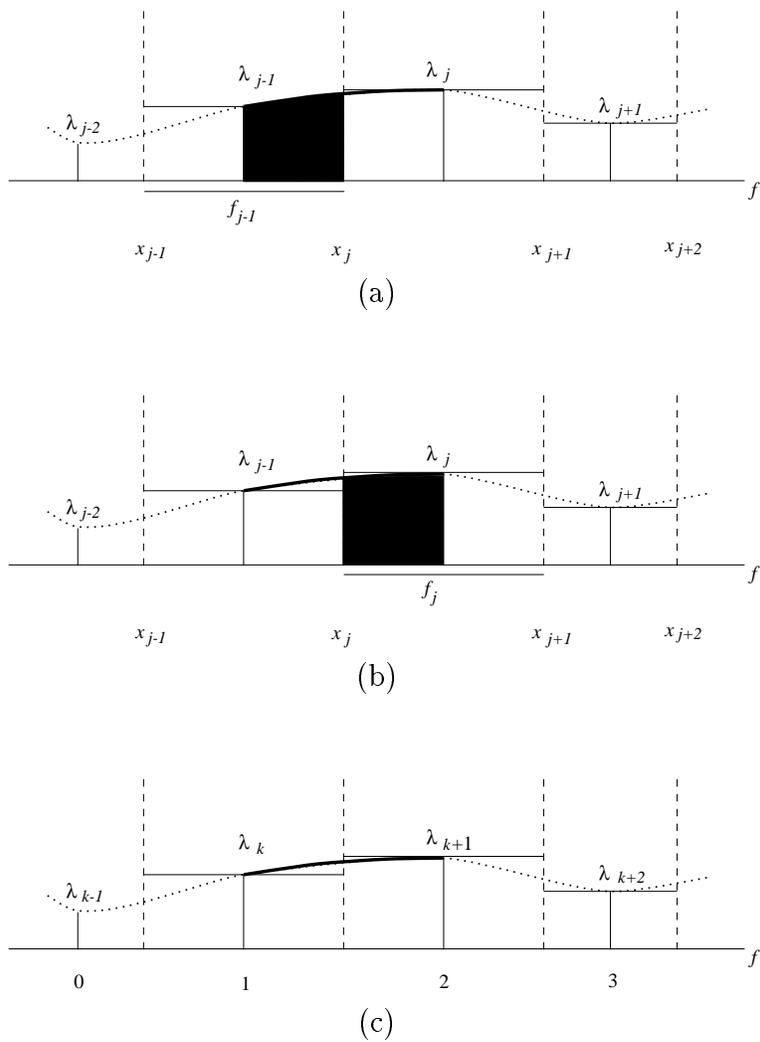


Figura 5.12: Segmento da função de tolerância: (a) pontos de contato que ocorrem na região preenchida para $t_c \geq 0.5$; (b) pontos de contato que ocorrem na região preenchida para $t_c < 0.5$; (c) segmento da função de tolerância com parâmetro $1 \leq t_\lambda < 2$.

5.5.1 Estimação do coeficiente de atrito

O atrito relativo ao ponto $f_j(t)$ pode ser calculado como uma função, tendo como parâmetro o valor da tolerância associada a esse ponto. Na seção anterior foi definida a forma de estimar um valor da tolerância $\lambda(t) = \gamma_k(t)$ relativa ao ponto $f_j(t)$.

A rugosidade, caracterizada pela tolerância do contorno, contribui para o coeficiente de atrito para cada ponto do contorno. Ela é associada a um parâmetro μ chamado *coeficiente de atrito* do contorno.

Em geral, o coeficiente μ pode ter valores superiores a zero, inclusive maiores que 1, mas comumente são considerados como sendo $0 \leq \mu \leq 1$. Como se deseja associar μ à rugosidade, se estabelece uma função μ em função de λ em cada ponto do contorno. Neste trabalho, por conveniência, se considera $0 \leq \lambda(t) \leq 1$. Na prática, tal restrição pode não ser cumprida. Assim, poderia ser necessário usar um escalar α (que pode ser a inversa do máximo das tolerâncias do sistema) de forma a reparametrizar $\lambda(t)$ para o intervalo unitário, $0 \leq \alpha\lambda(t) \leq 1$.

Considera-se a tolerância mínima (λ_{min}) como sendo a rugosidade de um contorno liso (aproximadamente lisa), e por isso se associa esta tolerância mínima ao valor mínimo de μ .

A função de coeficiente de fricção μ , nesta abordagem, é definida em função da tolerância λ como

$$\mu(\lambda_t) = \begin{cases} 0 & \text{se } \lambda(t) \leq \lambda_{min} \\ \lambda(t) - \lambda_{min} & \text{caso contrário.} \end{cases} \quad (5.6)$$

A esta função poderia, para maior realismo, se associar as outras variáveis que definem um coeficiente de atrito seco, tais como efeitos de desgaste modificando unicamente os vetores de que definem as perturbações relativas, etc.

Capítulo 6

Movimento dos corpos com interferências

Existem vários paradigmas para a geração do movimento dos corpos, em alguns casos complexos, mas em essência seguem uma mesma filosofia: calcular o estado do corpo em cada passo de tempo, baseado na formulação dinâmica ou cinemática de movimento.

O estado do corpo informa a sua posição e orientação. Com isto devem-se analisar os efeitos de interferência entre os corpos e para evitar o efeito *fantasma* (interpenetração). Técnicas de detecção de interferências entre os objetos em animação vêm-se aperfeiçoando cada vez, já que este consome a maior parte do tempo na simulação. Deve-se utilizar alguma estratégia para gerenciar recursos do computador (processo, espaço, tempo) e otimizar a complexidade do algoritmo na comparação geométrica somente dos objetos próximos, evitando assim testes desnecessários na comparação de objetos afastados um do outro.

Neste capítulo aborda-se a geração de movimento de corpos e detecções de suas interferências em vários níveis para, futuramente, evitar as interseções entre eles, abordados no Capítulo 7.

6.1 Dinâmica de movimento dos corpos

Quando um corpo se movimenta por translação e rotação, os dados de seu estado variam constantemente. A análise desse estado é feita em cada intervalo de tempo chamado passo da animação. Isto é, o estado de cada objeto é calculado em uma seqüência discreta de tempos. As situações destes corpos, nesses instantes de tempo, são mostradas via interfaces gráficas com características próprias do realismo de uma cena animada. A situação do corpo refere-se à sua posição e orientação no espaço de animação e, possivelmente, às mudanças de comportamento de alguns objetos devido à ocorrência de eventos de restrições, tais como colisões.

O cálculo de estados dos objetos em movimento é feito como uma evolução sequencial do sistema (evolução síncrona). Existe outra abordagem de cálculo de estados em paralelo (evolução assíncrona), tratado em [Celes98], em que o estado de cada corpo evolui independente dos outros. Ambas as abordagens são utilizadas para gerar movimento dos corpos e acompanhar sua evolução para a detecção de interferência entre os objetos.

6.1.1 Estado de corpo em movimento

Um corpo, em um instante t , tem um estado $\mathbf{Y}(t)$ composto pela posição $\mathbf{x}(t)$ de seu centro de massa no espaço, a orientação de seu sistema de coordenada local $\mathbf{R}(t)$, velocidade linear $\mathbf{v}(t)$ de seu centro de massa, e sua velocidade angular $\mathbf{w}(t)$. Assim, é frequente ver o estado de um corpo como

$$\mathbf{Y}(t) = \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{R}(t) \\ \mathbf{v}(t) \\ \mathbf{w}(t) \end{pmatrix}. \quad (6.1)$$

Em muitos casos, quando os corpos estão em movimento dinâmico, aconselha-se, por conveniência, que o elemento de estado $\mathbf{v}(t)$ seja substituído pelo momento linear $\mathbf{P}(t)$, e $\mathbf{w}(t)$ pelo momento angular $\mathbf{L}(t)$. Eles descrevem as informações das velocidades linear e angular de uma forma natural [Rivera96, Baraff92]. Com estas

observações, o estado de um corpo em movimento dinâmico será denotado como

$$\mathbf{Y}(t) = \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{R}(t) \\ \mathbf{P}(t) \\ \mathbf{L}(t) \end{pmatrix}. \quad (6.2)$$

A evolução de um corpo no tempo é determinada por uma equação diferencial, baseada na formulação dinâmica de Newton-Euler, como

$$\begin{aligned} \frac{d}{dt}\mathbf{Y}(t) &= \frac{d}{dt} \begin{pmatrix} \mathbf{x}(t) \\ \mathbf{R}(t) \\ \mathbf{P}(t) \\ \mathbf{L}(t) \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{v}(t) \\ \boldsymbol{\omega}(t) \times \mathbf{R}(t) \\ \mathbf{F}(t) \\ \mathbf{T}(t) \end{pmatrix}, \end{aligned} \quad (6.3)$$

onde $\mathbf{F}(t)$ e $\mathbf{T}(t)$ são a força e torque resultante atuando sobre o corpo, respectivamente.

A relação das grandezas da variação do estado (para mais detalhes, ver [Rivera96]) são dados como:

$$\begin{aligned} \mathbf{v}(t) &= \mathbf{P}(t)/m, \\ \mathbf{F}(t) &= \frac{d}{dt}\mathbf{P}(t), \\ \mathbf{T}(t) &= \frac{d}{dt}\mathbf{L}(t), \end{aligned}$$

onde m é a massa do corpo,

O momento angular total $\mathbf{L}(t)$ de um corpo rígido está relacionado com a inércia $\mathbf{I}(t)$ do corpo através da equação

$$\mathbf{L}(t) = \mathbf{I}(t).\boldsymbol{\omega}(t).$$

A equação diferencial (6.3) pode ser integrada, em cada passo de tempo, utilizando-se métodos numéricos para resolver equações diferenciais ordinárias (ODE), por exem-

plo o método de Euler, de Runge-Kutta ou algum outro método híbrido, dependendo da exatidão de valor desejado. Neste trabalho usa-se o método de Euler.

6.1.2 Evolução síncrona de movimento

Os estados de todos os corpos no instante t formam o estado do sistema $\mathbf{Y}(t)$, que registra a evolução de todos os objetos até o instante t . Dado o estado inicial $\mathbf{Y}(t_0)$, todo o sistema evolui, em um passo de tempo t_{step} , para o próximo estado que é registrado. Se não houver nenhuma interferência espacial entre os objetos, no intervalo $(t_0, t_0 + t_{step}]$, o novo estado adquirido pelo sistema é $\mathbf{Y}(t_0 + t_{step})$, que passa a ser estado inicial para o próximo passo de integração.

Se a interferência é detectada em $t_0 + t_{step}$, o estado inicial $\mathbf{Y}(t_0)$ é recuperado. Então, deverá se procurar um instante t_c , de forma $t_0 < t_c < t_0 + t_{step}$, usando-se alguma técnica de convergência de $\{t_i\}_{i \in I_c}$, instantes intermediários, para t_c , de forma que os estados intermediários $\{\mathbf{Y}(t_i)\}_{i \in I_c}$ sejam situações próximas ao de contato $\mathbf{Y}(t_c)$. O maior interesse neste processo é que I_c tenha o menor número de elementos possíveis para uma convergência rápida e robusta em t_c . Uma vez determinado o estado $\mathbf{Y}(t_c)$, são tratadas as interferências respectivas, que depois será completada a iteração deste último estado de contato até $t_0 + t_{step}$.

Como já foi observado, quando existe uma interferência espacial, o passo de tempo para atualizar os estados dos corpos pode ser menor que normal, já que este deve ser adaptado de t_0 para t_c , instante em que ocorre a interferência, mas não pode ser maior que t_{step} , a não ser que seja utilizada uma estratégia em que t_{step} seja adaptativo dependendo da média da rapidez de movimento dos corpos. Este critério traz algumas desvantagens neste paradigma, porque em objetos com movimentos muito lentos (t_{step} pequeno) faz com que a variação entre os frames seja pouco significativa, e t_{step} grande pode produzir efeito de grandes variações entre objetos que se movimentam rapidamente, então outra alternativa poderia ser passo de tempo adaptado, que depende do comportamento dos objetos em animação, abordagem utilizado por Celes [Celes98].

6.2 Detecção de interferências

São três os níveis de detecção de interferências analisados no presente trabalho: alto nível ou aproximado, nível de aproximação por hierarquias, e níveis em detalhe.

Alguns trabalhos, como proposto por Hubbard ([Hubbard95]), enfocam a análise de interferência em dois níveis: grosseira e refinada. A grosseira analisa a interseção de hiper-paralelepípedos formados pelas dimensões do objeto em movimento no espaço e tempo. A refinada verifica interferências pela interseção entre as esferas envolventes em uma hierarquia, seguindo uma estrutura de árvore esfera.

A abordagem de árvore de caixas orientadas permite uma comparação rápida em tempo constante pela interseção das caixas de um mesmo nível de hierarquia, já que existem duas caixas a comparar, sendo um máximo de quatro combinações, em cada nível da árvore, enquanto na árvore de esfera este número varia de dois a oito. Embora a verificação pela interseção de duas esferas seja muito mais simples do que das caixas, o número de caixas envolventes em cada nó é mínimo porque a árvore MOBBtree é binária.

O nível grosseiro de verificação de interferências entre os n objetos tem o objetivo minimizar o número de comparações entre eles em cada passo de tempo. Esta fase deve ser uma espécie de selecionador de possíveis objetos que possam estar em interferências. Um enfoque para isto pode ser o envelope espaço temporal abordado em detalhe em [Hubbard95], mas que finalmente se converte na verificação de interseção de $n \times n$ hiper-paralelepípedos, sendo os cálculos de interseção de elementos de cubos. A técnica utilizada por Baraff [Baraff92], que consiste em criar uma lista de pares de objetos próximos, utilizando-se a técnica de varredura [deBerg+97, PrepSham85], é uma boa opção para esta fase.

Dados m pares de objetos escolhidos dos n , pelo nível grosseiro, é verificada a interferência das caixas envolventes de cada um dos m pares, seguindo uma estrutura de árvore binária, até um nível dado de resolução do objeto. Uma vez confirmada a interferência, são determinados os segmentos de possíveis contatos, controlando os intervalos de perturbações desses segmentos, analiticamente por interseções.

6.2.1 Detecção de interferências de alto nível

O objetivo desta etapa é reduzir o número de pares de objetos cuja interferência deve ser verificada. Sem esta etapa, seria necessário fazer $O(n^2)$ dessas verificações. No entanto, a maior parte destes pares são de objetos que não estão sequer próximos.

Baraff [Baraff92] usa um algoritmo de ordenação e seleção de objetos que se comparam em $O(n+k)$, onde k indica o número de possíveis interseções dos lados de caixas envolventes, em relação a cada eixo de coordenadas. Cada caixa envolvendo um objeto é um paralelepípedo de arestas alinhadas aos eixos de coordenadas do sistema (conhecido como caixa *isotética*). A construção das caixas *isotéticas* em cada passo de tempo faz que o algoritmo seja um pouco lento. Critério análogo será utilizado, neste trabalho, sem a necessidade de construir caixas isotéticas adicionais; será usada a caixa de origem da árvore de caixas envolventes de cada objeto projetado nos eixos coordenados. Com este fim, é definido o que se entende por caixas próximas.

Definição 6.1 *Duas caixas orientadas B_1 e B_2 são consideradas próximas se os segmentos de suas respectivas projeções I_1 e I_2 sobre todos os eixos coordenados se intersectam.*

Duas caixas isotéticas, C_1 e C_2 , se intersectam se as projeções de seus respectivos lados sobre os eixos coordenados se intersectam [PrepSham85]. Projetando a caixa orientada B_i sobre os eixos coordenados, obtém-se os intervalos $\{I_i^j\}_{j=1,2}$ que constituem os lados de uma caixa isotética C_i . Nesta vez, o segmento $I_i^j = [b_i^j, e_i^j]$, é definido por

$$b_i^j = p_i^j - r_i^j \quad \text{e} \quad e_i^j = p_i^j + r_i^j. \quad (6.4)$$

O valor p_i^j é a projeção do centro da caixa B_i sobre j , e r_i^j é o raio do segmento projetado no mesmo eixo. A Figura 6.1 mostra as caixas orientadas com os segmentos projetados sobre os eixos coordenados.

A projeção de uma caixa sobre os eixos de coordenadas é feita considerando as devidas transformações (rotação e deslocamento) dos elementos vetoriais da caixa (eixos e centro da caixa) que estão no sistema local (SRO) para o sistema universal (SRU), já que cada caixa relativa a um objeto está expressada no SRO do objeto.

Dois intervalos $I_q^j = [b_q^j, e_q^j]$ e $I_r^j = [b_r^j, e_r^j]$ se intersectam, se:

$$b_r^j \leq b_q^j \leq e_r^j \quad (6.5)$$

ou

$$b_q^j \leq e_r^j \leq e_q^j. \quad (6.6)$$

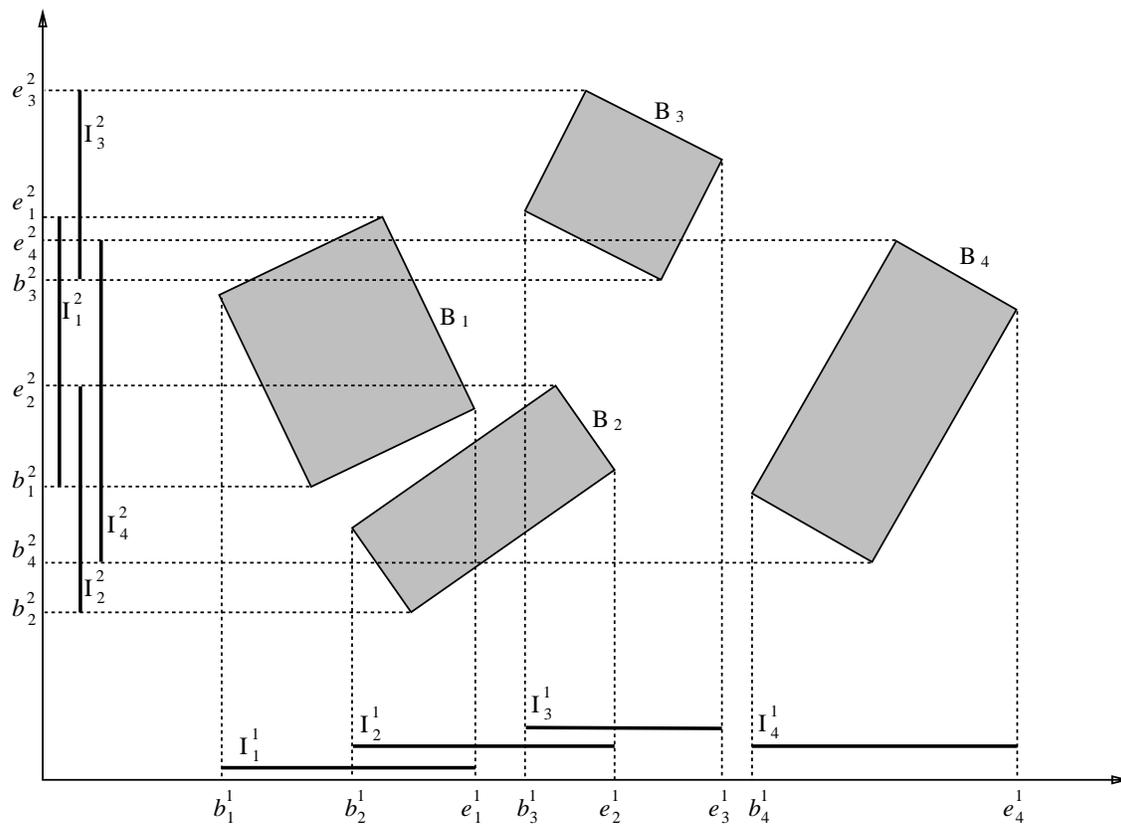


Figura 6.1: *Projeção de caixas, envolvendo o objeto, sobre os eixos coordenados.*

Algoritmo geral. O objetivo principal deste algoritmo é se obter uma lista de pares de objetos próximos com elementos (O_{k_i}, O_{r_i}) , para $i = 1, \dots, k$. Para isto, os processos são realizados na sequência a seguir:

- Projecção das caixas orientadas, $\{B_i\}_{i=1,\dots,n}$, sobre cada eixo coordenado x^j . O vetor de projecções, V^j para $j = 1, 2$, contém os $2n$ extremos dos intervalos de projecção $\{b_i^j, e_i^j\}_{i=1,\dots,n}$.
- Ordenação dos vetores $V^j, j = 1, 2$, em forma ascendente.
- Varredura do vetor V^1 , correspondente ao eixo x , de esquerda para a direita. Se $V^1[i]$ corresponde a um extremo b_k^1 , então o intervalo I_k^1 intersecta cada um dos intervalos ($\{I_{s_r}^1\}_{r=1,\dots,s}$) da lista ativa. Neste caso, deve-se verificar se os intervalos de projecções de B_k sobre os outros eixos também intersectam com seus equivalentes de B_{s_r} . Caso afirmativo, o par de objetos (O_k, O_{s_r}) são próximos, então deve-se adicionar à lista de pares próximos. Finalmente, se adiciona I_k^1 na lista ativa. Caso $V^1[i]$ corresponda a um extremo e_k^1 , o intervalo I_k^1 deve ser eliminado da lista ativa.

No algoritmo geral, pode-se observar que a projecção dos n objetos nos eixos coordenados é feita linearmente, demandando um tempo $O(n)$. A ordenação, utilizando-se um *QuickSort*, pode ser feita, em média, em $O(n \log n)$. A varredura é linear, $O(n)$, mas as operações de adição e eliminação dos elementos da lista ativa podem demandar um tempo $O(s \log s) = O(s \log n)$, porque $s \leq \binom{n}{2} = O(n^2)$ para s elementos ativos. Este processo pode ser linear se a lista ativa for administrada como um vetor de n posições, de forma que o elemento ativo seja indicado por um inteiro $r > -1$ que indica a posição do próximo elemento ativo. A adição de um par de objetos na lista de objetos próximos é constante. Sendo assim, o custo total da varredura $O(n + s)$, e o custo do algoritmo geral $O(n \log n + s)$.

O algoritmo geral se aplicará uma única vez, no começo da animação; nos passos de tempo seguintes, se utilizará um algoritmo mais simplificado, devido ao tipo de comportamento dos n objetos na animação. Em passos de tempo consecutivos, os n objetos não variam muito de posição e de orientação, mantendo desta forma uma coerência geométrica. Este comportamento vai ser refletido nos vetores de projecção V^j , já que um vetor V^j de um passo anterior vai ter poucas variações respeito ao corrente. Desta forma, o vetor V^j de projecção é *quase* ordenado. Por isso, o método de ordenação por inserção *Insertion Sort* pode ser utilizado eficientemente.

A filosofia do algoritmo de ordenação por inserção é ir trocando posições de elementos enquanto este for menor, mantendo os elementos da esquerda, cada vez, orde-

nados ascendentemente. Neste processo, poderá se observar que as trocas de posições entre elementos $V^j[i]$ e $V^j[i + 1]$, correspondentes a b_r e e_s (não necessariamente nessa ordem), vão gerar uma operação de adição e eliminação de pares (O_r, O_s) na lista de objetos próximos, dependendo do caso. Assim, o algoritmo geral pode ter uma nova versão mais simplificada e com menor custo, chamado algoritmo por coerência, detalhado a seguir.

Algoritmo por coerência. O objetivo deste algoritmo, de modo similar ao algoritmo geral, é fornecer a lista de pares de objetos próximos, mas operando sobre ela com adição e eliminação dos elementos pares, da forma seguinte:

- Projecção das caixas orientadas, $\{B_i\}_{i=1,\dots,n}$, sobre cada eixo coordenado; como no algoritmo geral, mas atualizando as posições de b_i e e_i de V^j com valores atuais.
- Ordenação dos vetores V^j , $j = 1, 2$, em forma ascendente, utilizando-se o método *Insertion Sort*; registrando-se, em caso de acontecer um evento b_r e e_s , os seguintes:
 - Ao ordenar V^1 , se $b_r \leq e_s$ verificar, como no algoritmo geral, a interseção dos segmentos respectivos nos outros eixos coordenados, correspondentes aos objetos O_r e O_s . Caso afirmativo, adicionar (O_r, O_s) na lista de objetos próximos. Se $b_r \leq e_s$ for falso, então eliminar o par (O_r, O_s) da lista de objetos próximos.
 - Ao ordenar V^j , $j > 1$, se $b_r \leq e_s$ é falso, então eliminar o par (O_r, O_s) da lista de objetos próximos, se ainda lá estiver.

O custo de processo no algoritmo por coerência é linear, já que o método *Insertion Sort*, para ordenar n dados em forma ascendente, utiliza um tempo $O(n)$ quando os dados são *quase* ordenados, e $O(n^2)$ quando os dados estão ordenados em forma decrescente [TeneAuge86]. Os dados aqui, neste trabalho, estão quase ordenados em forma ascendente, por isso o processo será feito em tempo linear. Outro aspecto importante notado neste algoritmo é que não se utiliza a lista ativa; desta forma, não serão necessárias as operações sobre a lista ativa que não existe explicitamente, mas aparecem as operações de adição e eliminação na lista de objetos próximos. O custo

de operações sobre a lista de objetos próximos pode ser minimizado com o uso de tabelas. O custo geral deste algoritmo será $O(n + k)$, onde k é o número de pares de adição e eliminação na lista de objetos próximos.

Para minimizar o custo de operações sobre a lista de objetos próximos, esta será administrada como um vetor de h elementos ($h \ll n$), de forma que a posição r do vetor aponta a uma mini-lista de pares (O_s, O_t) , tal que $r = (s + t) \bmod (h)$ (módulo h de $s + t$).

Existem outras abordagens para detectar interseções de segmentos por varredura em $O(n(\log n + s))$ utilizando-se princípios da árvore de intervalos [deBerg+97, PrepSham85, Wood85]. Outro algoritmo eficiente, proposto por Balaban [Balaban95], que utiliza um tempo de processo $O(n \log n + s)$ pode ser generalizado para resolver o problema inicial.

6.2.2 Interferências em hierarquia

Uma verificação rápida de interseção de caixas envolventes, utiliza o princípio da existência de um plano separador entre duas caixas retangulares, utilizado no método de testemunha de Baraff [Baraff92, Rivera96] e adaptado por Gottschalk et al. [GoLiMa96] para OBBtree, cuja abordagem é seguida neste trabalho, usando-se MOBBtree.

O critério consiste em, dadas duas caixas envolventes (A e B) de um nível correspondentes a um objeto cada uma, projetá-las sobre um plano separador e verificar a interseção das projeções no plano como se fosse em uma única dimensão (Vide Figura 6.2). A projeção de uma caixa retangular sempre vai ser um segmento (hexágono, em 3D, podendo tomar-se um círculo que o contém) com centro na projeção do centro da caixa.

Para efetuar a projeção das caixas A e B sobre uma linha, os elementos vetoriais da caixa (eixos e posição do centro) devem ser transformados para o sistema SRU, já que esses elementos estão expressos no SRO do respectivo objeto.

Duas caixas não intersectam se existe um plano separador entre elas (ver detalhes em [Rivera96]). O plano pode ser definido por um lado da caixa (em 3D, por uma face da caixa, ou por uma aresta de uma caixa e um vértice da outra caixa). Sendo assim, o plano de projeção das caixas é perpendicular ao plano de separação. Se as

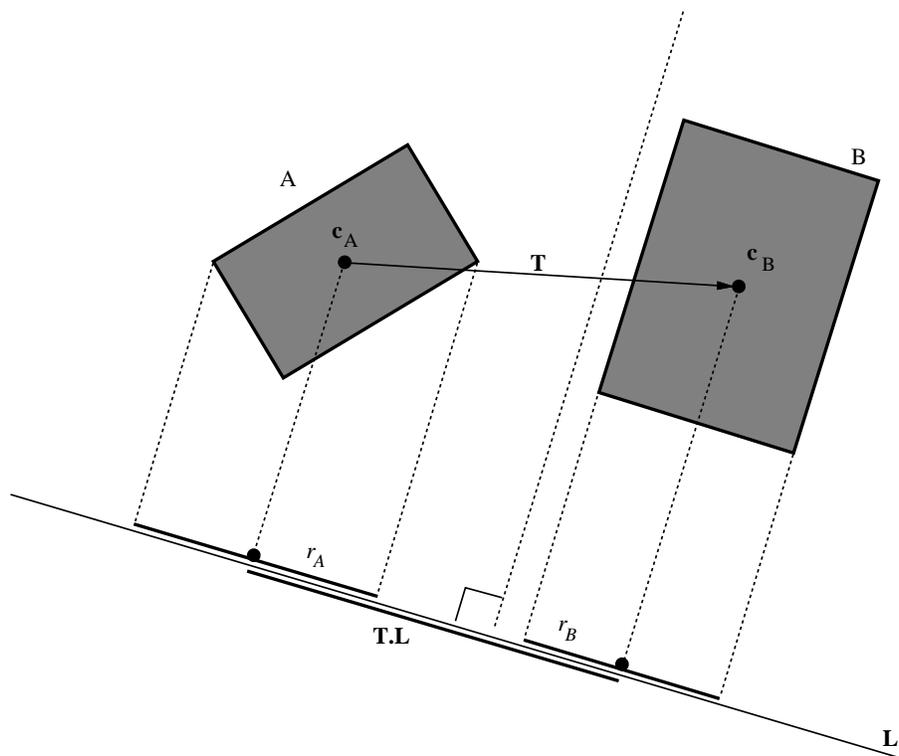


Figura 6.2: *Projeção de caixas A e B sobre a reta L: comprimento T.L maior que a semi soma dos segmentos projetados indica que as caixas não intersectam.*

projeções das caixas sobre o plano de projeção não se intersectam, então as caixas são disjuntas.

A verificação da existência de plano, neste caso reta, de separação é simples. Cada caixa tem dois únicos lados orientados (em 3D, três faces, e três arestas orientadas). Estes elementos podem fornecer 4 possíveis retas (em 3D, 15 planos). Se duas caixas se intersectam, será necessário o teste de todas as possíveis retas de separação.

Para o teste de interseção, deve-se projetar o centro das caixas sobre a reta, e calcular o raio dos respectivos intervalos. A reta de projeção L é perpendicular a um eixo de uma das caixas. A reta L pode ser escolhida de modo a passar pelo ponto central da caixa A, por conveniência, para efeitos de cálculos. O vetor de separação

entre os centros das duas caixas, $\mathbf{T} = \mathbf{c}_B - \mathbf{c}_A$, também é projetado sobre \mathbf{L} , sendo sua magnitude $L = |\mathbf{T} \cdot \mathbf{L}|$.

Os raios dos segmentos projetados (semi-segmentos), da caixa A, são obtidos como

$$r_A = \max\{r_A^j = |\mathbf{v}_A^j \cdot \mathbf{L}|, \text{ tal que } \mathbf{v}_A^j = (-1)^j \cdot l_A^1 \cdot \mathbf{e}_A^1 + l_A^2 \cdot \mathbf{e}_A^2, \text{ para } j = 1, 2\}, \quad (6.7)$$

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.

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} . Considerou-se que seja reportada a projeção da metade dos vetores indicadores dos vértices adjacentes, neste caso 2, porque uma caixa retangular tem esses vetores simétricos dois a dois. Outro aspecto que se considera, nestes cálculos, é que o semi-segmento de cada caixa é suficiente para o cômputo da interseção dos segmentos projetados, uma vez que se tem a posição da projeção do centro da cada caixa, e a projeção da distância dos centros das duas caixas. Então, as caixas A e B são consideradas disjuntas se satisfazem a condição

$$L > r_A + r_B. \quad (6.8)$$

Como são quatro eixos que definem possíveis linhas de separação entre A e B, o teste da interseção envolve quatro casos. Se a condição (6.8) 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 (no caso de 3D, no entanto, seriam 15 planos a se testar).

Uma vez detectada a interseção das caixas A e B, são verificados os pares de filhos destas caixas, passando para um nível superior, seguindo os mesmos critérios aqui descritos. Este processo é seguido, até o nível desejado, enquanto existir interseção das caixas comparadas. Se não houver interseção entre A e B em algum nível, se considera que os pedaços cobertos pelas respectivas caixas não se intersectam. Se duas caixas finais (ou folhas) se intersectam, este par de caixas é registrado na lista de caixas intersectantes para seu futuro teste de interseção em detalhe, analiticamente, tratado na próxima seção.

Quando duas árvores são de tamanhos diferentes, a raiz da árvore menor é recursivamente testada com as caixas de nós filhos da árvore maior, enquanto forem intersectantes, até encontrar a subárvore do mesmo tamanho que a árvore de tamanho menor. Depois, as comparações por interseção são continuadas normalmente como explicado acima. O segmento de código a seguir mostra a seqüência seguida para testes de interferência entre caixas A e B de duas árvores, cujas alturas são *alt_a* e *alt_b* respectivamente:

```

interferencia_caixas (alt_a, alt_b, A, B)
{
  // alt_a: altura da subarvore A
  // alt_b: altura da subarvore B
  se intersectam_caixas (A, B), então
    se alt_a e alt_b são menores desejados, então
      A e B intersectam. Adicionar o par A e B na lista de pares intersectados lst.
    senão
      se alt_a > alt_b, então
        // nivelar a subarvore A com a subarvore B
        interferencia_caixa (alt_a - 1, alt_b, filho esquerdo de A, B)
        interferencia_caixa (alt_a - 1, alt_b, filho direito de A, B)
      senão
        se alt_a < alt_b, então
          // nivelar a subarvore B com a subarvore A
          interferencia_caixa (alt_a, alt_b - 1, A, filho esquerdo de B)
          interferencia_caixa (alt_a, alt_b - 1, A, filho direito de B)
        senão
          // ambas subaroves são do mesmo tamanho
          interferencia_caixa (alt_a - 1, alt_b - 1, filho esquerdo de A, filho esquerdo de B)
          interferencia_caixa (alt_a - 1, alt_b - 1, filho esquerdo de A, filho direito de B)
          interferencia_caixa (alt_a - 1, alt_b - 1, filho direito de A, filho esquerdo de B)
          interferencia_caixa (alt_a - 1, alt_b - 1, filho direito de A, filho direito de B)
        fse
      fse
    fse
  fse
}

```

Ao finalizar a comparação das duas árvores, se não fora registrada nenhuma caixa folha intersectando, então não há interferência entre os dois objetos das árvores. Caso haja caixas folha registradas, significa que é possível que exista interseção dos dois objetos, o qual será constatado pela verificação de interferência em detalhe. Na Figura 6.3 mostra-se o estado de dois objetos em dois instantes, o inicial e o instante quando se detectam as interferências das caixas básicas dos dois objetos.

6.2.3 Interferências em detalhe

Cada par de caixas básicas em interseção podem incluir segmentos de superfícies que estejam em três situações: intersectados, em contato, ou afastados. A sequência de ações do algoritmo de verificação de interferências em detalhe, Figura 6.4, mostra as três situações.

Dois segmentos estão intersectados quando existe no mínimo um ponto de interseção entre os segmentos; ou seja, quando existe uma parte de um segmento no interior do outro. Esta situação indica que deverá se voltar a uma nova iteração da dinâmica com um passo menor, porque nesta iteração foi dado um passo que permitiu a interpenetração dos objetos. por isto, não será mais necessário continuar com a verificação dos outros pares de caixas em interferência. Com o primeiro indicador de interpenetração de segmentos deverá se ignorar todos os estados obtidos nesta iteração dinâmica.

Dois segmentos estão em contato quando existe um ponto sobre cada segmento, de forma que são geometricamente próximos, com uma tolerância λ permitida. Isto é, os pontos $f(t_f)$ e $g(t_g)$, dos segmentos f e g , respectivamente, estão suficientemente próximos quando o vetor $\mathbf{d}(t_f, t_g) = f(t_f) - g(t_g)$ tem a direção da normal comum aos dois os segmentos e, além disso, $0 \leq \|\mathbf{d}(t_f, t_g)\| \leq \lambda$. Com isto, neste trabalho, somente são considerados corpos em contato quando exista interseção tangencial entre ambos os corpos, ou quando eles estão bem próximos com uma tolerância permitida, incorporada em cada caixa.

Segmentos afastados são aqueles que não se intersectam, nem estão em contato. Deste modo, quando ocorrer a situação em contato ou afastado, o processo de detecção de pontos de contatos continua com os outros pares de caixas em interferência. Só na situação de interseção de segmentos é truncada a seqüência de verificação de contato,

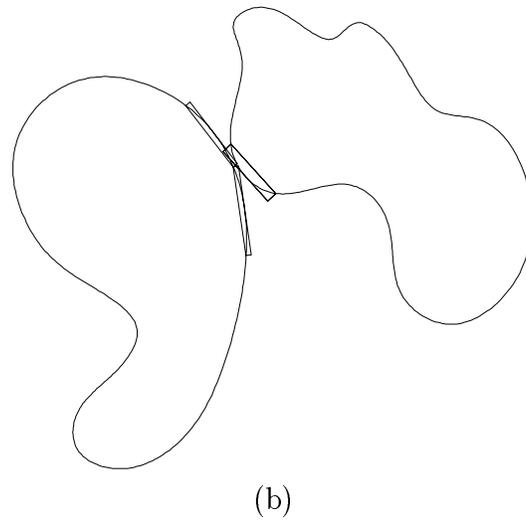
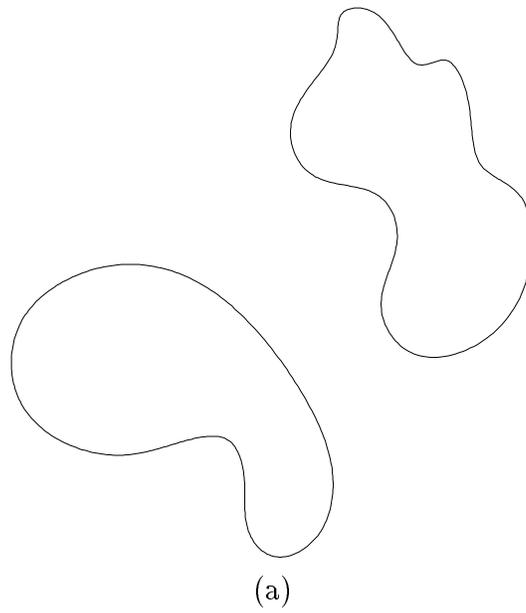


Figura 6.3: *Estados de dois objetos na cena de animação: (a) momento inicial; (b) momento em que se detectam interferências entre as primeiras caixas básicas.*

```

inteiro contatos(lst_caixas, lst_pts)
{
  . . .
  n_contatos = 0
  enquanto lst_caixas não nulo, fazer
  . . . // verificar a interseção dos segmentos  $f_a$  e  $f_b$  de lst_caixas
  se  $f_a$  e  $f_b$  intersectam, então
  . . . //  $f_a$  e  $f_b$  intersectam
  n_contatos = -1 // indicador da existencia de interpenetração
  parar // existe interpenetração, interrompe a verificação
  senão
  . . . // verificar contato tolerado entre  $f_a$  e  $f_b$  de lst_caixas
  se existe contato, então
  . . . //  $f_a$  e  $f_b$  em contato, armazenar ponto de contato em lst_pts
  n_contatos = n_contatos + 1 // adiciona numero de contatos
  senão
  . . . // segmentos  $f_a$  e  $f_b$  estão afastado
  fse
  lst_caixas = seguinte de lst_caixas
  fse
  enquanto
  retorna n_contatos // ha n_contatos contatos (0 = todos afastados).
}

```

Figura 6.4: Trecho de código contendo lógica de detecção de contatos em detalhe.

para re-iniciar uma nova iteração dinâmica como explicado anteriormente.

A aproximação aos pontos de contato, quando é detectada uma interpenetração, é feita via a técnica de *biseção*. Esta técnica consiste na partição binária do passo de tempo: diminuir o tamanho quando existe uma interpenetração, e incrementar em metade do passo anterior quando todos são afastados. Assim, a convergência para o instante de contato ocasiona uma seqüência de passos de tempos em *zig zag* (para maiores detalhes desta técnica, ver [PrTeVe+92]).

Quando ocorrer a situação de contato, cada par de pontos de contatos, declarados como tais, são armazenados numa lista de pontos de contato (*lst_pts* na Figura 6.4). Esses pontos vão ser tratados para evitar interpenetrações nas iterações dinâmicas futuras.

6.2.4 Detecção de pontos de contato

A classificação de situações entre cada par de segmentos pode ser feita de várias formas. Uma forma básica é através de subdivisão binária; outra forma é via formulação analítica.

A subdivisão binária, para este problema, consiste em construir dinamicamente caixas envolventes de sub-segmentos, segundo a filosofia de árvore binária, até se considerar que elas estão em contato ou separadas. Similar abordagem é seguida por Gleicher e Kass [GleiKass92], mas criando paralelepípedos de arestas paralelas aos eixos do sistema de referência do universo.

A abordagem pela formulação analítica consiste em prosseguir analiticamente pelas técnicas de interseção de curvas e superfícies curvas, dadas em maior ênfase em geometria e modelagem de sólidos [Hoffmann89]. Estas técnicas foram usadas, entre outros, por Baraff [Baraff90], Min Lin [Lin94] e Snyder [Snyder95] para resolver casos similares, mas com várias formas de resolução.

A robustez do método de solução depende do tamanho do pedaço da superfície considerada onde possivelmente se encontra a solução. Esses pedaços são analisados localmente a interseção ou proximidade de atributos das superfícies, via iteração numérica. Para isto é necessário isolar os possíveis pontos de contato e interseções, o que não é uma coisa simples.

Na colisão, não é necessário determinar a interseção exata das superfícies, mas sim pontos próximos a pontos tangenciais que podem ser aproximados por projeções. Verificar que objetos estão interpenetrados, basta demonstrar que existe uma interseção de seus componentes.

A abordagem analítica é baseada na formulação do problema em termos de resolução de sistemas de equações algébricas para se obter soluções locais.

Das duas abordagens, a primeira é lenta em convergir para o resultado requerido. Ela fornece resultados não exatos, porque se considera a média de uma vizinhança de pontos. A sucessiva construção de caixas orientadas para cada sub-segmento é mais cara que o cálculo iterativo do método analítico. Neste trabalho, optou-se pelo método analítico para resolver o problema de contatos do nível segmentos, combinando com métodos de subdivisão dos intervalos de seus domínios.

Método numérico de equações homogêneas

A formulação analítica do problema de colisão é dada em forma de equações não lineares homogêneas,

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} h_1(\mathbf{x}) \\ \dots \\ h_m(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} 0 \\ \dots \\ 0 \end{bmatrix} = \mathbf{0}, \quad \text{para } \mathbf{x} = (t_1, t_2, \dots, t_n). \quad (6.9)$$

Um sistema de equações não lineares homogêneas, da forma (6.9), não pode, em geral, ser resolvido analiticamente. Somente pode ser resolvido, com um grau de aproximação bastante alto, numericamente, por iteração partindo de um ponto inicial \mathbf{x}_0 . Existem métodos robustos que podem convergir a bons resultados, mas que têm suas particularidades. Por isto, não existe um bom método geral para resolver sistemas de mais de uma equação não-linear [PrTeVe+92].

O método multidimensional de Newton-Raphson é o mais simples e robusto quando o valor inicial, \mathbf{x}_0 , está próximo ao valor desejado. No entanto, quando este está afastado, a tendência a extremos relativos pode desvirtuar a convergência. Para superar essa patologia existem formas de controlar o comportamento do processo de interação na busca de raízes [PrTeVe+92]. Com uma escolha de um ponto inicial de possível solução, \mathbf{x}_0 , próximo à solução desejada $\tilde{\mathbf{x}}$, a convergência pelo método de Newton-Raphson é rápida.

O método numérico de Newton consiste em levar (6.9) para um sistema linear de equações homogêneas em cada iteração, gerando-se uma seqüência de valores \mathbf{x}_i de forma a convergir a um resultado que seria a raiz $\tilde{\mathbf{x}}$ de (6.9).

A formulação de Newton-Raphson para sistema de equações não-lineares, na i -ésima iteração, é da forma

$$\mathbf{H}(\mathbf{x}_i) = \mathbf{H}(\mathbf{x}_{i-1}) + \frac{\partial \mathbf{H}}{\partial \mathbf{x}}(\mathbf{x}_{i-1}) (\mathbf{x}_i - \mathbf{x}_{i-1}), \quad (6.10)$$

onde $\frac{\partial \mathbf{H}}{\partial \mathbf{x}}(\mathbf{x}_{i-1})$ é o Jacobiano avaliado em \mathbf{x}_{i-1} . O sistema de equações lineares homogêneas, em cada iteração de Newton-Raphson, é dado por

$$\frac{\partial \mathbf{H}}{\partial \mathbf{x}}(\mathbf{x}_{i-1}) (\mathbf{x}_i - \mathbf{x}_{i-1}) + \mathbf{H}(\mathbf{x}_{i-1}) = \mathbf{0}, \quad (6.11)$$

o que pode ser resolvido, dependendo da característica e a complexidade da matriz envolvida no sistema, por métodos como *decomposição LU*, *SVD - decomposição do valor singular*, ou qualquer outro método eficiente do método numérico [PrTeVe+92], sendo a incógnita $\mathbf{u}_i = \mathbf{x}_i - \mathbf{x}_{i-1}$.

Formulação de sistemas de equações

As três situações em que podem estar dois segmentos, f_r e g_s , são expressas através de sistemas de equações não-lineares homogêneas, de forma similar à expressão geral (6.9), como:

1. Interseção entre f_r e g_s é definida por

$$\mathbf{H}(t_f, t_g) = f_r(t_f) - g_s(t_g) = \mathbf{0}. \quad (6.12)$$

Este sistema consiste de 2 equações com 2 incógnitas, e é de grau 3 porque cada segmento é B-spline cúbico.

2. Contato entre f_r e g_s que pode ser *contato tangencial* ou *contato tolerado*.

- contato tangencial é uma interseção entre os dois segmentos em seus pontos tangentes. Por isto, à expressão (6.12) é adicionada a condição de tangentes anti-paralelas:

$$H_3(t_f, t_g) = f'_r(t_f) \times g'_s(t_g) = 0. \quad (6.13)$$

Um sistema de ponto tangencial está composto por 3 equações (2 de (6.12) e 1 de (6.13)) e 2 incógnitas.

- contato tolerado é quando existem pontos próximos entre os dois segmentos, de forma que o vetor distância $\mathbf{d}(t_f, t_g) = f(t_f) - g(t_g)$ satisfaz a condição $0 < \|\mathbf{d}(t_f, t_g)\| \leq \lambda$, para $0 \leq t_f, t_g < 1$. As equações não lineares para este tipo de pontos são dadas por:

$$\left. \begin{aligned} H_1(t_f, t_g) &= (f(t_f) - g(t_g)) \cdot f'(t_f) = 0 \\ H_2(t_f, t_g) &= (f(t_f) - g(t_g)) \cdot g'(t_g) = 0. \end{aligned} \right| \quad (6.14)$$

Este sistema é composto por 2 equações e 2 incógnitas. Cada equação formulada tem um grau $2n - 1$ (obtido depois de operar o produto da diferença entre as duas superfícies e a derivada respectiva), e neste caso é 5.

Observa-se que as equações não lineares, neste caso, tem duas variáveis e duas equações, excepto para o ponto tangencial. Para uniformizar o número de equações para o uso de um método resolvidor de equações lineares, similar a (6.11), pode-se pensar, em geral, em um sistema de duas equações e duas incógnitas. No caso de ponto tangencial, primeiro deve cumprir a equação (6.12), depois verificar a condição de suas tangentes, expressão (6.13). Na prática, o contato tangencial nunca vai acontecer, porque a probabilidade de um contato tangencial é mínima, mas de qualquer forma ele é considerado.

A matriz Jacobiana, $\frac{\partial \mathbf{H}}{\partial \mathbf{x}} = \mathbf{J}$, para um sistema de 2 equações e 2 incógnitas é de ordem 2×2 , sendo em geral

$$\mathbf{J} = \begin{bmatrix} \frac{\partial H_1}{\partial t_f} & \frac{\partial H_1}{\partial t_g} \\ \frac{\partial H_2}{\partial t_f} & \frac{\partial H_2}{\partial t_g} \end{bmatrix}.$$

Assim, o Jacobiano para interseções, (6.12), é dado por

$$\mathbf{J} = \begin{bmatrix} f'_x(t_f) & -g'_x(t_g) \\ f'_y(t_f) & -g'_y(t_g) \end{bmatrix}.$$

e o Jacobiano para contato tolerado, (6.14), tem a forma

$$\mathbf{J} = \begin{bmatrix} \|f'(t_f)\|^2 + (f(t_f) - g(t_g)) \cdot f''(t_f) & -g'(t_g) \cdot f'(t_f) \\ g'(t_g) \cdot f'(t_f) & -\|g'(t_g)\|^2 + (f(t_f) - g(t_g)) \cdot g''(t_g) \end{bmatrix}.$$

Com jacobiano de ordem 2×2 , um sistema de equações lineares tipo (6.11), pode ser resolvido sem recorrer aos sistemas resolvidores mencionados acima, já que um método simples de resolver duas equações e duas incógnitas será suficiente.

A tolerância λ , usada para a definição de pontos de contato, varia para cada par de segmentos. Como introduzida no Capítulo 2 e detalhada no Capítulo 5, cada segmento de superfície do objeto tem uma tolerância que depende da distribuição de rugosidade associada a ele. Assim, se λ_f e λ_g são as tolerâncias associadas aos segmentos f_r e g_s , respectivamente, a tolerância para a análise de interferência em detalhe desse par de segmentos é dada por

$$\lambda = \lambda_f + \lambda_g. \quad (6.15)$$

Assim, se os segmentos f e g forem próximos nos pontos $f(t_f)$ e $g(t_g)$, a condição para contato tolerado é

$$\|f(t_f) - g(t_g)\| \leq \lambda.$$

Aproximação de valores mínimos

A resolução de um sistema de equações não-lineares, $\mathbf{H}(\mathbf{x}) = \mathbf{0}$, por métodos numéricos requer um ponto \mathbf{x}_0 inicial de aproximação, para convergir iterativamente ao valor desejado (se existir) $\tilde{\mathbf{x}}$.

O método de Newton é eminentemente local, e não pode ser usado com um \mathbf{x}_0 inicial arbitrário. Para explorar a eficiência deste método, requer-se o uso de outros métodos que proporcionem uma primeira aproximação \mathbf{x}_0 da solução desejada.

Analisando os casos de segmentos que podem envolver cada caixa básica, como se explica na Seção 3.3, observa-se que eles são segmentos curvos localmente convexos ou com um ponto de inflexão, de modo que subdivididos no ponto médio do intervalo de seus argumentos, $0 \leq t_f, t_g < 1$, cada subsegmento é localmente convexo. Com este critério, para cada par de subsegmentos, a função $\mathbf{H}(t_f, t_g)$ pode ter de zero a dois mínimos locais, sendo, possivelmente, um deles o zero da função, exceto no caso degenerado de contatos tolerados, que podem ter infinitos pontos (mas para interseção é suficiente detetar o primeiro zero da função).

Para escolher o par de subsegmentos em que serão aplicados o critério de convergência por Newton-Raphson, usa-se uma heurística de seleção: ordenar os pares de subsegmentos pela distância entre seus pontos médios, de menor a maior. Como podem existir dois pontos mínimos locais, a distância mínima não é suficiente para uma convergência para o resultado. Por isto, o método de Newton é aplicado em pares de subsegmentos selecionados pelas distâncias de menor a maior, no caso de não convergir nas anteriores, sendo \mathbf{x}_0 o ponto médio do sub-intervalo relativo ao par de subsegmentos escolhido. Com este método, na prática a convergência ao resultado acontece, com muita frequência, no primeiro par de subsegmentos. O pior caso acontece quando não existe solução de $\mathbf{H}(\mathbf{x}) = \mathbf{0}$ no intervalo $0 \leq t_f, t_g < 1$.

A Figura 6.5 mostra os quatro sub-intervalos do domínio de $\mathbf{H}(\mathbf{x})$ cujos centros, (t_f^0, t_g^0) , (t_f^0, t_g^1) , (t_f^1, t_g^0) e (t_f^1, t_g^1) , são os possíveis pontos próximos ao ponto de solução de $\mathbf{H}(t_f, t_g)$.

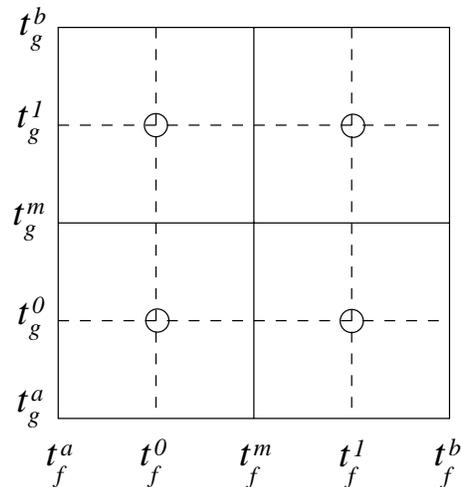


Figura 6.5: Intervalo de domínio de cada segmento é dividido em quatro sub-intervalos.

6.3 Performance de detecção de interferências

A performance de OBBtree para objetos complexos de superfícies poligonais é discutida por Gottschalk et al. [GoLiMa96], mostrando uma eficiência superior a todos os outros métodos existentes. O método MOBBtree herda de OBBtree a eficiência nas comparações pela interseção das caixas, porque o método de detecção de interferência das caixas em MOBBtree é similar ao usado em OBBtree, considerando ainda que neste trabalho o método formulado é unicamente no espaço bidimensional, o que torna ainda mais rápida a detecção de interferências entre as caixas orientadas da MOBBtree.

Para ilustrar a performance do método com caixas bidimensionais, formula-se um exemplo de quatro situações de três objetos de 512 segmentos cada um. As situações são mostradas na Figura 6.6: objetos afastados, duas situações em que existem alguns segmentos 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 MOBBtrees foi efetuada na estação *Sun SPARCstation 20* com *Solaris*. O tempo medido é o mínimo tempo em segundos de vários experimentos, já que eles podem variar de um experimento a outro devido ao sistema de multiprocessamento em que trabalha a máquina usada.

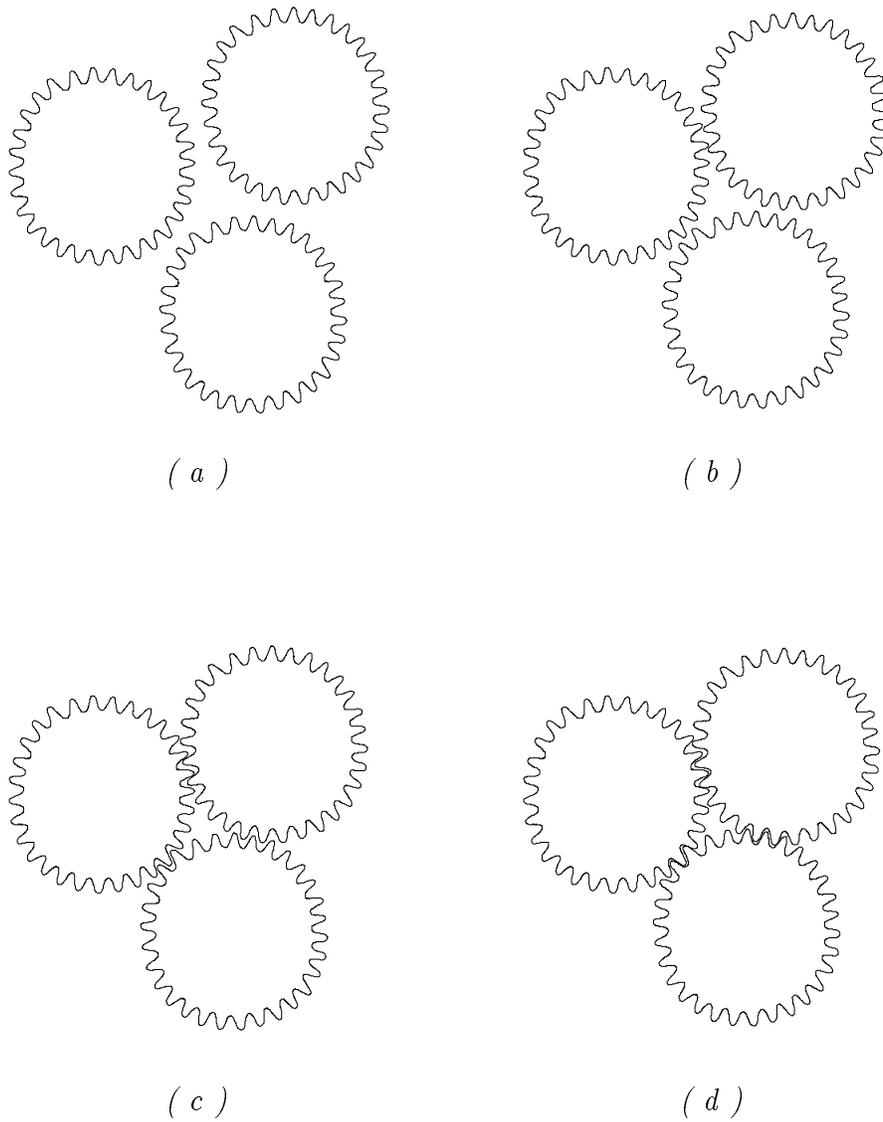


Figura 6.6: Situação de três objetos com 512 segmentos cada um: (a) afastados; (b) contato entre dois objetos; (c) contato entre todos os objetos; (d) maior número de contatos entre todos os objetos.

A Figura 6.7 mostra uma tabela com os resultados obtidos, que mostram 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.

Em suma, a detecção de interferências com o uso de MOBBtree é bastante eficiente e possibilita animação em tempo real, mesmo com um número considerável de objetos.

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)	0.007	343
(d)	0.01	459

Figura 6.7: Resultados da execução de quatro situações de três objetos de 512 segmentos cada um, segundo a Figura 6.6.

Capítulo 7

Tratamento de pontos de contatos

O tratamento de colisões e contatos tem como objetivo principal evitar as interpenetrações entre objetos, modificando apropriadamente a tendência de seus movimentos [MoorWhil88, Hahn88, Baraff89, Baraff92, Kamat93, Mirtich96]. Nesta etapa de animação, deve-se levar em conta a elasticidade e fricção das superfícies em contato, que ocasionam perdas de energia cinética no momento de colisão.

As colisões causam descontinuidade nas velocidades dos corpos, enquanto os contatos contínuos causam a descontinuidade nas acelerações, devido às forças que atuam em cada corpo. As descontinuidades de velocidades é estabelecida por impulsos, de acordo com as leis de conservação de momento linear e angular dos corpos. As descontinuidades de acelerações, na visão dinâmica, envolvem adição ou eliminação de restrições de acelerações nos objetos. O paradigma clássico de conservação de momentos é bastante aceito no cálculo de impulsos. Existe ainda outro método de calcular impulsos exatos considerando as variações das velocidades durante o impacto, abordagem seguida por Mirtich [Mirtich96].

A seqüência tradicional para resolver colisões e contatos consiste em, primeiro, tratar as colisões no nível de velocidades e, depois, alterar as acelerações, porque o cálculo das acelerações geralmente requer que as alterações geradas pelas variações das velocidades sejam conhecidas [Hahn88, Cremer89, Baraff92].

As colisões podem ser resolvidas por métodos *analíticos* ou de *penalização*. O método analítico se baseia na conservação dos momentos linear e angular de cada corpo envolvido na colisão; este método é tradicional e mais barato na resolução de

colisões [MoorWhil88]. Já o método de penalização aproveita os princípios da força de mola que, inseridas entre os corpos no ponto de colisão, evita as interpenetrações.

Os contatos contínuos podem ser tratados de três formas, através do método de *penalização*, do método *dinâmico* e do método baseado em *impulsos*. O método de penalização, do mesmo modo que no tratamento de colisões, consiste em inserir temporariamente forças de mola entre os corpos no ponto de contato; estes, com suas propriedades, mantêm os corpos sem interpenetrações [MoorWhil88, Witkin94, TerWit88, Amirou92, Christo94]. O método dinâmico consiste em se formular os contatos mediante as leis da dinâmica. As equações resultantes são, de modo geral, quadráticas, devido à presença de restrições de acelerações, e sua solução envolve sofisticadas técnicas numéricas de resoluções de equações de restrições simultâneas, paradigma proposto inicialmente por Lötstedt [Lotstedt81], e enfatizado em [Baraff92, Cremer89]. O método baseado em impulso, também é formulado mediante as leis físicas, mas evita a restrição de acelerações devido à hipótese de que não existe um repouso absoluto de corpos em contato contínuo; em vez disto, eles sofrem uma série de *micro-colisões*. Portanto, por este critério, todo tipo de contato deve ser resolvido como sendo uma colisão, ou seja através de impulsos [MirtCan94, Mirtich96]. Este paradigma foi proposto inicialmente por Hahn [Hahn88].

Tanto o método dinâmico quanto o método baseado em impulso são métodos analíticos; a diferença está no tratamento de contatos contínuos, como explicado anteriormente, sendo mais simples o método baseado em impulsos. Como o método dinâmico requer técnicas de complementaridade para resolver sistemas de equações diferenciais de segundo ordem com restrições, ele não é apropriado para animação em tempo real, porque a resolução de métodos de complementaridade é computacionalmente cara. Por outro lado, o método fornece resultados físicos exatos.

Neste capítulo são classificados analiticamente os pontos de contato, formulam-se e resolvem-se as equações de impulso baseado no princípio de conservação de momentos, e finalmente tratam-se os contatos como sendo micro-colisões.

7.1 Pontos de contato

Chama-se ponto de contato entre dois objetos A e B a um ponto comum \mathbf{p} definindo um ponto de colisão ou um ponto de contato contínuo; para maior detalhe ver [Baraff92, Rivera96]. Um ponto de colisão, também conhecido como ponto de impacto, é um contato instantâneo, de tempo de duração apenas infinitesimal, enquanto o ponto de contato contínuo tem uma duração finita. Estes contatos são classificados por suas velocidades relativas, definidas como

$$\mathbf{v}_{rel}(t) = \dot{\mathbf{p}}_a(t) - \dot{\mathbf{p}}_b(t), \quad (7.1)$$

onde $\dot{\mathbf{p}}_a(t)$ e $\dot{\mathbf{p}}_b(t)$ são as velocidades de \mathbf{p} em relação aos corpos A e B, respectivamente. A velocidade de um ponto (\mathbf{p}_a relativa a A, e \mathbf{p}_b a B) é definida como

$$\dot{\mathbf{p}}_a(t) = \mathbf{v}_a(t) + \boldsymbol{\omega}_a(t) \times \mathbf{r}_a, \quad \text{com} \quad \mathbf{r}_a = \mathbf{p}_a(t) - \mathbf{X}_a(t), \quad (7.2)$$

onde \mathbf{v}_a , $\boldsymbol{\omega}_a$ e \mathbf{X}_a são velocidade linear, velocidade angular e posição de centro de massa do corpo A, respectivamente.

Se considera temporariamente um sistema de coordenadas no ponto \mathbf{p} , cujo eixo vertical é definido pelo vetor normal $\hat{\mathbf{n}}$ (vetor normal de B para A no ponto \mathbf{p}). O eixo horizontal é definido pelo vetor tangente de contato.

A velocidade relativa, \mathbf{v}_{rel} , pode ser decomposta em *velocidade relativa normal* \mathbf{v}_{rel_n} , com vetor unitário coincidente com à normal $\hat{\mathbf{n}}$, e *velocidade relativa tangencial* \mathbf{v}_{rel_t} que está no plano tangente com vetor unitário $\hat{\mathbf{t}}$, tal como mostra a Figura 7.1. Assim,

$$\mathbf{v}_{rel_n} = v_{rel_n} \hat{\mathbf{n}} \quad (7.3)$$

$$\mathbf{v}_{rel_t} = v_{rel_t} \hat{\mathbf{t}} = \mathbf{v}_{rel} - \mathbf{v}_{rel_n}, \quad (7.4)$$

onde os escalares v_{rel_n} e v_{rel_t} são obtidos pelo produto escalar do vetor \mathbf{v}_{rel} com $\hat{\mathbf{n}}$ e $\hat{\mathbf{t}}$, respectivamente:

$$v_{rel_n} = \mathbf{v}_{rel} \cdot \hat{\mathbf{n}} \quad \text{e} \quad v_{rel_t} = \mathbf{v}_{rel} \cdot \hat{\mathbf{t}}.$$

O vetor unitário da velocidade relativa tangencial, $\hat{\mathbf{t}}$, é definido como

$$\hat{\mathbf{t}} = \frac{\mathbf{v}_{rel} - \mathbf{v}_{rel_n}}{\|\mathbf{v}_{rel} - \mathbf{v}_{rel_n}\|} \quad (7.5)$$

No caso tridimensional, o vetor unitário $\hat{\mathbf{t}}$ estará no plano tangente, portanto deverá se decompor em dois vetores unitários ortogonais $\hat{\mathbf{i}}$ e $\hat{\mathbf{j}}$, ver [Baraff92, Mirtich96, Hahn88]. Mas no caso bidimensional, $\hat{\mathbf{t}}$ é indicador suficiente da direção tangente de contato¹.

Considerando o escalar v_{rel_n} da velocidade relativa normal, são classificados os pontos de contatos em:

- $v_{rel_n} > 0$: separação, indica que A e B estão se separando no ponto \mathbf{p} ,
- $v_{rel_n} < 0$: colisão, indica que A e B estão se interpenetrando, e
- $v_{rel_n} = 0$: contínuo, indica que A e B estão em contato contínuo em \mathbf{p} .

7.2 Impacto baseado em impulso

Quando existe uma colisão todas as forças externas são ignoradas; somente são consideradas as *forças impulsivas*. A força impulsiva, chamada também de força de colisão, (\mathbf{f}_c) atua entre os corpos em contato, gerada como uma reação de um corpo pela ação do outro no ponto de contato. Esta força é composta por uma força de restrição de interpenetração (\mathbf{f}_n) e uma força de fricção (\mathbf{f}_t),

$$\mathbf{f}_c = \mathbf{f}_n + \mathbf{f}_t. \quad (7.6)$$

A força de restrição não produz trabalho, e é uma força na direção da normal $\hat{\mathbf{n}}$. Já força de fricção \mathbf{f}_t tem direção $\hat{\mathbf{t}}$, e produz trabalho por dissipação de energia cinética devido à fricção de contato.

A força de contato, atuando nos pontos de contato durante o intervalo de tempo infinitesimal, Δt , produz a variação do momento linear e angular dos corpos que colidem. A ação da força impulsiva durante Δt é o impulso, definida como

$$\mathbf{J} = \int_{\Delta t} \mathbf{f}_c dt. \quad (7.7)$$

O impulso \mathbf{J} é o parâmetro vetorial que faz mudança das velocidades dos corpos para evitar a interpenetração. A conservação de momento linear e angular estabelece

¹Caso a colisão for estritamente vertical no frame local de colisão, a velocidade tangencial poderia ser nula, por tanto $\hat{\mathbf{t}} = \mathbf{0}$

a relação de impulso com as velocidades dos dois corpos em colisão, A e B, como

$$\begin{aligned} m_a \mathbf{v}_a^+ &= m_a \mathbf{v}_a + \mathbf{J} \\ m_b \mathbf{v}_b^+ &= m_b \mathbf{v}_b - \mathbf{J} \end{aligned} \quad (7.8)$$

$$\begin{aligned} \mathbf{I}_a \boldsymbol{\omega}_a^+ &= \mathbf{I}_a \boldsymbol{\omega}_a + \mathbf{r}_1 \times \mathbf{J} \\ \mathbf{I}_b \boldsymbol{\omega}_b^+ &= \mathbf{I}_b \boldsymbol{\omega}_b - \mathbf{r}_2 \times \mathbf{J} \end{aligned} \quad (7.9)$$

\mathbf{I}_i é o momento de inércia do corpo i , \mathbf{v}_i e $\boldsymbol{\omega}_i$ são velocidades linear e angular antes da colisão, e \mathbf{v}_i^+ e $\boldsymbol{\omega}_i^+$ são as velocidades linear e angular de i após a colisão. Na realidade, os elementos que permanecem constantes durante a colisão são \mathbf{I}_i , a massa m_i e a posição de contato \mathbf{r}_i ([MacMilla36, Mirtich96]). O termo $\mathbf{r}_i \times \mathbf{J}$ é o *torque impulsivo* gerado por \mathbf{J} no corpo i .

Para resolver as equações de conservação de momentos, é necessário considerar as condições de contato: restituição de colisão e fricção de contato.

O *princípio empírico de Newton*, que vincula a condição de restituição com a velocidade, é dado por

$$v_{rel-n}^+ = -\varepsilon \cdot v_{rel-n}, \quad (7.10)$$

onde v_{rel-n} é a componente normal de velocidade relativa definida em (7.3). Como as duas superfícies podem ter coeficientes de restituição diferentes, para um ponto de colisão, se considera $\varepsilon = \min \{\varepsilon_a, \varepsilon_b\}$.

A condição de fricção afeta somente a velocidade relativa tangencial, $\mathbf{v}_{rel-t} = v_{rel-t} \hat{\mathbf{t}}$. Assim, a força tangencial, \mathbf{f}_t , atua no sentido contrário de \mathbf{v}_{rel-t} , de modo que

$$\mathbf{f}_t = -f_t \hat{\mathbf{t}}. \quad (7.11)$$

A *lei de Coulomb*, que estabelece a fricção como uma relação empírica entre a força normal e a força tangencial, é dada por:

$$\begin{aligned} \text{se } v_{rel-t} \neq 0 &\implies f_t = \mu \cdot f_n, \\ \text{se } v_{rel-t} = 0 &\implies f_t < \mu \cdot f_n. \end{aligned} \quad (7.12)$$

A constante μ é o coeficiente de fricção, que depende do material e rugosidade da superfície do corpo em contato. Neste caso, se considera $\mu = \max \{\mu_a, \mu_b\}$, já que os coeficientes de fricção entre duas superfícies podem ser diferentes.

A primeira condição da regra de Coulomb, (7.12) ($v_{rel-t} \neq 0$) indica que a força de fricção é *dinâmica*, enquanto a segunda ($v_{rel-t} = 0$) indica que a força de fricção é *estática*. Tipicamente, o coeficiente de fricção estática (μ_{stat}) é maior que o coeficiente de fricção dinâmica (μ_{dyna}), mas para os efeitos de cálculos de forças de contato, neste trabalho, não se fará distinção entre ambas as fricções. Além disso μ não depende da velocidade relativa tangencial.

Neste trabalho, o coeficiente de fricção, μ , para cada ponto de contato é obtido a partir das perturbações da superfície do objeto, definido na Seção 5.5, variando ao longo da superfície de cada corpo.

Para resolver um ponto de colisão com as formulações acima indicadas, é comum considerar as condições da regra de Coulomb, (7.12), uma por vez [MacMilla36, Hahn88, Kamat93, MoorWhil88]. Sendo assim, o ponto de colisão se resolve em dois passos:

- Primeiro passo, supõe-se que existe um atrito alto o suficiente, de forma a cumprir a segunda regra de Coulomb (fricção estática). Para isto será necessário que

$$v_{rel-t} = 0. \quad (7.13)$$

Com esta condição, o sistema de 10 equações (8 de (7.8) e (7.9), 1 de (7.10) e 1 da última suposição) e 10 incógnitas (\mathbf{v}_a^+ , \mathbf{v}_b^+ , \mathbf{w}_a^+ , \mathbf{w}_b^+ e \mathbf{J}) é tipicamente resolvido por um método tal como o de eliminação de Gauss-Jordan ou decomposição LU [PrTeVe+92, ChaCan85]. Com os resultados obtidos, é verificada a restrição estática:

$$J_t < \mu \cdot J_n.$$

Se esta for satisfeito, os resultados obtidos são corretos; caso contrário, a suposição (7.13) é errada, e portanto, $v_{rel-t} \neq 0$. Deve-se passar para o segundo passo.

- Segundo passo, deve ser cumprida a condição de fricção dinâmica,

$$J_t = \mu \cdot J_n. \quad (7.14)$$

Ou seja, $\mathbf{J} \cdot \hat{\mathbf{t}} = \mu \mathbf{J} \cdot \hat{\mathbf{n}}$. Com esta condição deve-se resolver novamente o sistema das 10 equações, já que $v_{rel-t} \neq 0$ afeta o valor de J_n da fricção dinâmica.

A resolução do sistema de equações acima definidas para um único ponto, pode ser simplificada para uma simples expressão de avaliação do valor de \mathbf{J} ; deste modo, evita-se o uso de métodos resolvidores de sistemas de equações lineares. Uma vez obtido o valor de \mathbf{J} , as velocidades após a colisão são computadas diretamente por (7.8) e (7.9). Na próxima seção aborda-se a simplificação das equações de impactos.

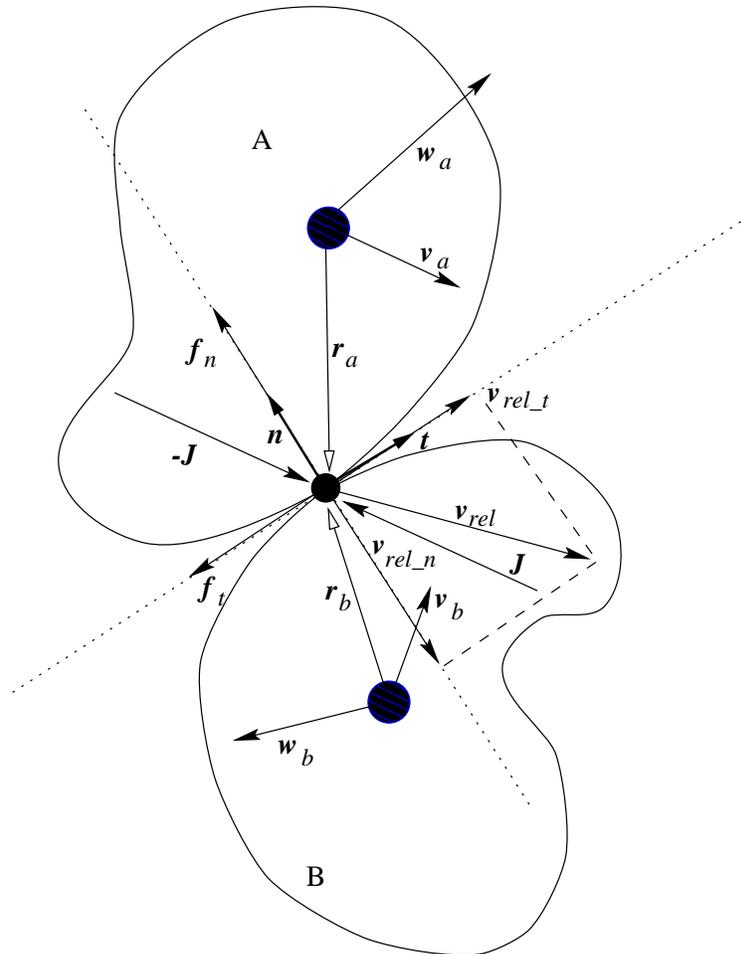


Figura 7.1: Colisão entre A e B, indicando alguns elementos que são considerados na análise de colisão. O eixo vertical do sistema de referência de colisão é alinhado com o vetor normal.

7.2.1 Impulso em um ponto de contato

O sistema de equações para colisões sem atrito, conformada por (7.8), (7.9) e (7.10), pode ser reduzido para uma única expressão que computa J_n . A redução formal deste caso é apresentada em [Rivera96, Baraff94]. Baseado nesse método, formula-se o cálculo de impulso para uma colisão, combinando as equações estabelecidas na seção anterior.

O vetor de impulso \mathbf{J} é expresso em termos dos vetores unitários normal e tangencial, $\hat{\mathbf{n}}$ e $\hat{\mathbf{t}}$, e os módulos respectivos J_n e J_t , como

$$\mathbf{J} = J_n \hat{\mathbf{n}} - J_t \hat{\mathbf{t}}, \quad (7.15)$$

que representa a expressão após a integração de (7.6).

A velocidade do ponto \mathbf{p}_a após a colisão, combinando (7.2) com (7.8) e (7.9), é dada por

$$\begin{aligned} \dot{\mathbf{p}}_a^+ &= \dot{\mathbf{p}}_a + \frac{1}{m_a} \mathbf{J} + [\mathbf{I}_a^{-1} (\mathbf{r}_a \times \mathbf{J})] \times \mathbf{r}_a \\ &= \dot{\mathbf{p}}_a + \frac{1}{m_a} \mathbf{J} - (\tilde{\mathbf{r}}_a \mathbf{I}_a^{-1} \tilde{\mathbf{r}}_a) \mathbf{J} \\ &= \dot{\mathbf{p}}_a + \left[\frac{1}{m_a} \mathbf{1} - (\tilde{\mathbf{r}}_a \mathbf{I}_a^{-1} \tilde{\mathbf{r}}_a) \right] \mathbf{J}, \end{aligned}$$

onde $\tilde{\mathbf{r}}_a$ é a matriz antissimétrica dual do vetor $\mathbf{r}_a = (r_{a1}, r_{a2}, r_{a3})$ definida como

$$\tilde{\mathbf{r}}_a = \begin{bmatrix} 0 & -r_{a3} & r_{a2} \\ r_{a3} & 0 & -r_{a1} \\ -r_{a2} & r_{a1} & 0 \end{bmatrix},$$

e $\mathbf{1}$ é a matriz de identidade. O mesmo critério é seguido para \mathbf{p}_b , com a diferença de que \mathbf{J} atua sobre B no sentido contrário ao de sua atuação sobre A:

$$\dot{\mathbf{p}}_b^+ = \dot{\mathbf{p}}_b - \left[\frac{1}{m_b} \mathbf{1} - (\tilde{\mathbf{r}}_b \mathbf{I}_b^{-1} \tilde{\mathbf{r}}_b) \right] \mathbf{J}.$$

A velocidade relativa do ponto de contato, (7.1), é expressa combinando as expressões acima,

$$\begin{aligned} \mathbf{v}_{rel}^+ &= \dot{\mathbf{p}}_a^+ - \dot{\mathbf{p}}_b^+ \\ &= \dot{\mathbf{p}}_a - \dot{\mathbf{p}}_b + \left[\frac{1}{m_a} \mathbf{1} - (\tilde{\mathbf{r}}_a \mathbf{I}_a^{-1} \tilde{\mathbf{r}}_a) \right] \mathbf{J} + \left[\frac{1}{m_b} \mathbf{1} - (\tilde{\mathbf{r}}_b \mathbf{I}_b^{-1} \tilde{\mathbf{r}}_b) \right] \mathbf{J} \\ &= \mathbf{v}_{rel} + \left[\left(\frac{1}{m_a} + \frac{1}{m_b} \right) \mathbf{1} - (\tilde{\mathbf{r}}_a \mathbf{I}_a^{-1} \tilde{\mathbf{r}}_a + \tilde{\mathbf{r}}_b \mathbf{I}_b^{-1} \tilde{\mathbf{r}}_b) \right] \mathbf{J}. \end{aligned}$$

Consequentemente pode-se expressar a velocidade relativa de um ponto após a colisão como

$$\mathbf{v}_{rel}^+ = \mathbf{v}_{rel} + \mathbf{K} \cdot \mathbf{J}, \quad (7.16)$$

onde a matriz \mathbf{K} , chamada *matriz de colisão*, é definida como

$$\mathbf{K} = \left(\frac{1}{m_a} + \frac{1}{m_b} \right) \mathbf{1} - \left(\tilde{\mathbf{r}}_a \mathbf{I}_a^{-1} \tilde{\mathbf{r}}_a + \tilde{\mathbf{r}}_b \mathbf{I}_b^{-1} \tilde{\mathbf{r}}_b \right). \quad (7.17)$$

Esta matriz é fundamental no cálculo de colisões, por isso é necessário analisar suas propriedades.

Lema 7.1 *A matriz $\mathbf{S}_i = -\tilde{\mathbf{r}}_i \mathbf{I}_i^{-1} \tilde{\mathbf{r}}_i$ da expressão (7.17) é constante, simétrica e positiva semi-definida.*

Prova: \mathbf{S}_i é constante já que \mathbf{r}_i e \mathbf{I}_i^{-1} são constantes durante a colisão. Como \mathbf{I}_i^{-1} é simétrica e positiva definida, e $\tilde{\mathbf{r}}_i$ é antisimétrica, $\mathbf{S}_i^T = \mathbf{S}_i$, portanto \mathbf{S}_i é simétrico. Seja um vetor arbitrário \mathbf{x} de forma que

$$\mathbf{x}^T \mathbf{S}_i \mathbf{x} = -\mathbf{x}^T \tilde{\mathbf{r}}_i \mathbf{I}_i^{-1} \tilde{\mathbf{r}}_i \mathbf{x} = (\tilde{\mathbf{r}}_i \mathbf{x})^T \mathbf{I}_i^{-1} (\tilde{\mathbf{r}}_i \mathbf{x}) = (\mathbf{r}_i \times \mathbf{x})^T \mathbf{I}_i^{-1} (\mathbf{r}_i \times \mathbf{x}).$$

Desde que \mathbf{I}_i^{-1} é positiva definida, $(\mathbf{r}_i \times \mathbf{x})^T \mathbf{I}_i^{-1} (\mathbf{r}_i \times \mathbf{x})$ é não negativo, portanto \mathbf{S}_i é positivo semi-definido.

◇

Teorema 7.1 *A matriz de colisão \mathbf{K} , definida em (7.17), de dois corpos A e B em colisão, é constante, não singular, simétrica e positiva definida.*

Prova: Como os elementos massa m_i , tensor de inércia \mathbf{I}_i e posição de pontos de contato \mathbf{r}_i são constantes durante o intervalo de colisão, a matriz \mathbf{K} é constante. Para as outras propriedades, \mathbf{K} é a soma de três matrizes simétricas positivas definidas: produto de escalar por matriz identidade, que é simétrica positiva definida; $\mathbf{S}_a = -\tilde{\mathbf{r}}_a \mathbf{I}_a^{-1} \tilde{\mathbf{r}}_a$ e $\mathbf{S}_b = -\tilde{\mathbf{r}}_b \mathbf{I}_b^{-1} \tilde{\mathbf{r}}_b$. Segundo o lema anterior, \mathbf{S}_a e \mathbf{S}_b são constantes, simétricas e positivas semi-definidas, portanto \mathbf{K} , que é dada pela soma de três matrizes simétricas, das quais uma é positiva definida e duas são positiva semi-definidas, é simétrica positiva definida, que implica que ela é não singular.

◇

O Teorema 7.1 garante que \mathbf{K} é inversível, logo (7.16) pode ser reformulada como

$$\mathbf{J} = \mathbf{K}^{-1}(\mathbf{v}_{rel}^+ - \mathbf{v}_{rel}). \quad (7.18)$$

Teorema 7.2 *Em uma colisão com fricção, o impulso para verificação de fricção estática é dado por*

$$\mathbf{J} = -\mathbf{K}^{-1}(\mathbf{v}_{rel} + \varepsilon \cdot v_{rel-n} \cdot \hat{\mathbf{n}}).$$

Prova: A regra de Newton, (7.10), afeta somente a componente normal da velocidade relativa, ou seja,

$$\mathbf{v}_{rel}^+ = v_{rel-t}^+ \hat{\mathbf{t}} - \varepsilon \cdot v_{rel-n} \cdot \hat{\mathbf{n}},$$

que substituída em (7.18) seria

$$\begin{aligned} \mathbf{J} &= \mathbf{K}^{-1}(v_{rel-t}^+ \hat{\mathbf{t}} - \varepsilon \cdot v_{rel-n} \cdot \hat{\mathbf{n}} - \mathbf{v}_{rel}) \\ &= -\mathbf{K}^{-1}(\mathbf{v}_{rel} + \varepsilon \cdot v_{rel-n} \cdot \hat{\mathbf{n}}) + v_{rel-t}^+ \mathbf{K}^{-1} \cdot \hat{\mathbf{t}}. \end{aligned} \quad (7.19)$$

Na fricção estática $v_{rel-t}^+ = 0$; com isto (7.19) passa a ser uma simples expressão de avaliação de valores constantes. Assim

$$\mathbf{J} = -\mathbf{K}^{-1}(\mathbf{v}_{rel} + \varepsilon \cdot v_{rel-n} \cdot \hat{\mathbf{n}}). \quad (7.20)$$

◇

Se a colisão for de fricção estática, deve satisfazer $J_t < \mu J_n$. Caso contrário, a colisão tem fricção dinâmica satisfazendo $J_t = \mu J_n$.

A componente normal de impulso em uma colisão com fricção dinâmica é obtida a partir de (7.16), combinando com (7.15), e considerando a restrição negativa da componente normal da velocidade relativa. Como a velocidade relativa após a colisão é dada por

$$\mathbf{v}_{rel}^+ = \mathbf{v}_{rel} + \mathbf{K}(J_n \hat{\mathbf{n}} - J_t \hat{\mathbf{t}}),$$

a condição da fricção dinâmica ($J_t = \mu J_n > 0$) implica

$$\mathbf{v}_{rel}^+ = \mathbf{v}_{rel} + J_n(\mathbf{K} \cdot \hat{\mathbf{n}} - \mu \mathbf{K} \cdot \hat{\mathbf{t}}).$$

Multiplicando ambos os lados por $\hat{\mathbf{n}}$, e substituindo $v_{rel_n}^+ = -\varepsilon v_{rel_n}$, obtém-se

$$-\varepsilon v_{rel_n} = v_{rel_n} + J_n(\hat{\mathbf{n}} \cdot \mathbf{K} \cdot \hat{\mathbf{n}} - \mu \hat{\mathbf{n}} \cdot \mathbf{K} \cdot \hat{\mathbf{t}})$$

Como um contato é considerado colisão quando $v_{rel_n} < 0$, então $v_{rel_n}^+ = -\varepsilon v_{rel_n} > 0$.

Logo

$$v_{rel_n} + J_n(\hat{\mathbf{n}} \cdot \mathbf{K} \cdot \hat{\mathbf{n}} - \mu \hat{\mathbf{n}} \cdot \mathbf{K} \cdot \hat{\mathbf{t}}) > 0.$$

Isto quer dizer que

$$\mu \hat{\mathbf{n}} \cdot \mathbf{K} \cdot \hat{\mathbf{t}} - \hat{\mathbf{n}} \cdot \mathbf{K} \cdot \hat{\mathbf{n}} < v_{rel_n},$$

e, portanto, a componente normal do impulso de fricção dinâmica é dada por

$$J_n = \frac{(1 + \varepsilon)v_{rel_n}}{\mu \hat{\mathbf{n}} \cdot \mathbf{K} \cdot \hat{\mathbf{t}} - \hat{\mathbf{n}} \cdot \mathbf{K} \cdot \hat{\mathbf{n}}}.$$

Um caso particular da expressão do impulso de fricção dinâmica ocorre quando $\mu = 0$. Neste caso, a força de contato não tem componente de fricção, portanto não existe fricção na colisão (similar ao apresentado em [Rivera96]). Note que $\hat{\mathbf{n}} \cdot \mathbf{K} \cdot \hat{\mathbf{n}} > 0$, já que \mathbf{K} é simétrica e positiva definida. Tem-se, portanto, o seguinte teorema.

Teorema 7.3 *A componente normal de Impulso de uma colisão sem atritos é dada por*

$$J_n = -\frac{(1 + \varepsilon)v_{rel_n}}{\hat{\mathbf{n}} \cdot \mathbf{K} \cdot \hat{\mathbf{n}}}.$$

7.3 Contato contínuo

Dois objetos estão em contato contínuo em um ponto \mathbf{p} quando a componente normal de sua velocidade relativa é considerada zero (ou seja, $v_{rel_n} \approx 0$). Na abordagem baseada em restrições [Baraff92], a dinâmica deste tipo de contatos é analisada através de acelerações relativas. Na abordagem baseada em impulsos, não será necessário fazer uma análise ao nível de acelerações, já que se parte da hipótese de que cada par de objetos em contato contínuo estão como se estivessem em constantes vibrações mínimas; isto é, estão em constantes micro-colisões, seguindo uma trajetória balística

[Mirtich96, Hahn88]. A componente normal da velocidade relativa inicial, v_{rel_n} , é limitada por

$$-v_{rel_n} < \sqrt{2g\lambda}, \quad (7.21)$$

onde g é a aceleração da gravidade e λ é a tolerância de contato definida na Seção 5.1. Com este critério, a perturbação da componente normal da velocidade relativa é calculada em função da profundidade de penetração δ do ponto de contato na tolerância λ , como

$$v_{rel_n} = -\sqrt{2g\delta}.$$

A fórmula acima é deduzida da equação clássica da velocidade $v_f^2 = v_0^2 + 2g\lambda$, onde v_0 é a velocidade inicial do corpo. Aqui, v_0 é a velocidade relativa normal, v_{rel_n} , antes de tratamento de contato.

O coeficiente de restituição para contato contínuo, ε_c , também depende do grau de profundidade de penetração δ no intervalo de tolerância λ . Deve-se ter, necessariamente

$$0 \leq \delta \leq \lambda.$$

A Figura 7.2 mostra a relação de δ com λ para um ponto de contato.

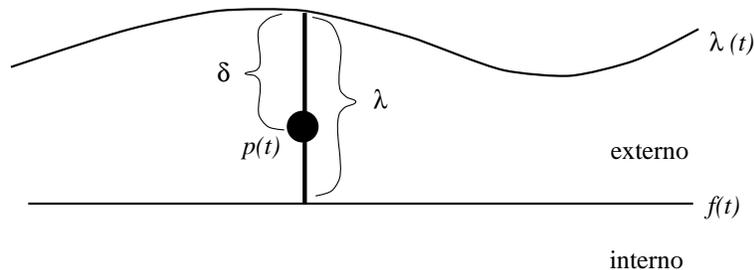


Figura 7.2: Um ponto de contato com o grau de penetração δ no intervalo de tolerância λ .

Se ε é o coeficiente de restituição normal, usada para análise de colisões, tem-se $0 \leq \varepsilon \leq 1$, e considera-se um coeficiente de restituição máximo ε_{max} , de forma que

$\varepsilon \leq \varepsilon_c(\delta) \leq \varepsilon_{max}$. O coeficiente de restituição $\varepsilon_c(\delta)$ adotado é dado por uma curva de segunda ordem com mínimo em $\delta = 0$:

$$\varepsilon_c(\delta) = \frac{\varepsilon_{max} - \varepsilon}{\lambda^2} \delta^2 + \varepsilon.$$

A Figura 7.3 mostra o comportamento da curva $\varepsilon_c(\delta)$ entre ε e ε_{max} ao longo da profundidade de penetração do ponto de contato no intervalo de tolerância λ . Assim, esta função dá maior restituição quando maior é o valor de δ .

Em vista de que λ_i não é constante em toda a superfície de um objeto, a restituição para contato contínuo para um par de objetos é dada por

$$\varepsilon_c = \max \{ \varepsilon_c(\delta_i), i = a, b \}.$$

O valor de ε_{max} pode ser a maior das constantes de restituições do sistema, mas um valor próximo a 1, por exemplo $\varepsilon_{max} = 0.9$, funciona adequadamente produzindo uma vibração mínima dos corpos para mantê-los em repouso.

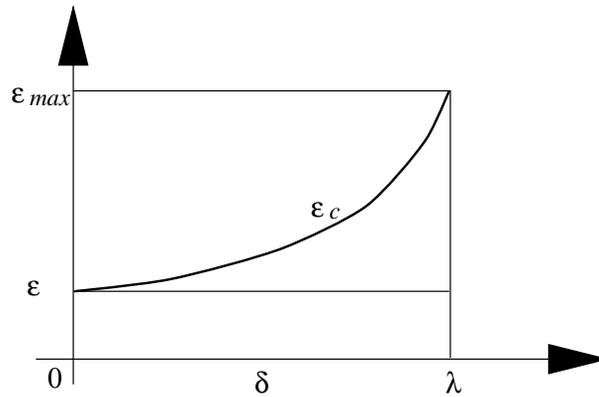


Figura 7.3: A função que define coeficiente de restituição para dinâmica de contato contínua. O valor de coeficiente de restituição depende do grau de penetração no intervalo de tolerância λ .

7.4 Exemplos de resultados obtidos de simulação dinâmica

Nesta seção são apresentados alguns exemplos de animação de objetos, onde os seus comportamentos dependem da complexidade dos atributos físicos de suas superfícies.

Comportamento de atritos

Na dinâmica entre dois corpos ocorrem uma série de eventos esperados. Em uma cena com vários corpos, cada par de corpos experimenta eventos similares. Por isto, seria difícil mostrar o comportamento de atrito de superfícies irregulares em uma cena de múltiplos objetos. Neste sentido, considerou-se conveniente, para o comportamento de atrito, uma cena composta de um corpo esférico e um bloco horizontal, Figura 7.4. Ao rolar a esfera sobre a superfície plana horizontal, detecta-se o atrito que varia em cada ponto de contato.

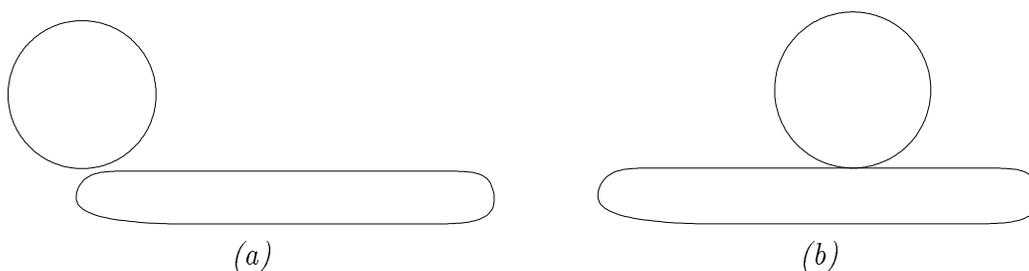


Figura 7.4: Um exemplo de dinâmica de contato para um objeto esférico que se movimenta sobre um objeto horizontal: (a) estado inicial; (b) estado em animação.

Na Figura 7.5 pode-se observar o comportamento do atrito de contato durante a movimentação do objeto esférico sobre o bloco horizontal em repouso, variando os tipos de rugosidades distribuídas nas suas superfícies. Na parte (a) da figura mostra-se o comportamento de atrito constante (similar ao método tradicional de se definir atrito constante em toda a superfície). As outras figuras mostram a variação de atrito em cada ocorrência de colisão, em diferentes pontos da superfície. Nessa figura, foi necessário unir os pontos de atrito de cada colisão, porque pontos separados não seriam perceptíveis para diferenciar suas tendências.

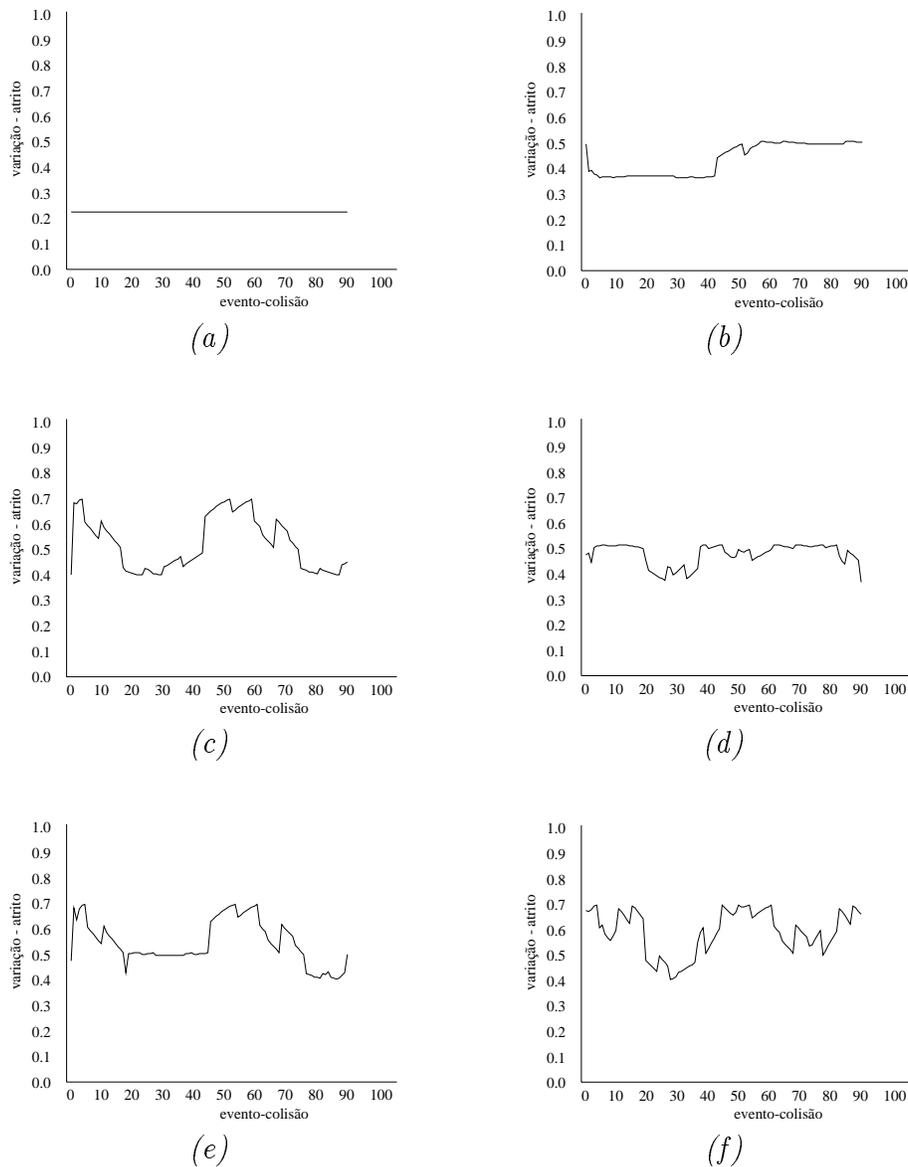


Figura 7.5: *Comportamento do atrito em evento de colisão do corpo esférico em movimento sobre a superfície horizontal em repouso, com tipos de rugosidades variadas, para esfera - bloco horizontal: (a) tipo_1 - tipo_1; (b) tipo_1 - tipo_2; (c) tipo_1 - tipo_3; (d) tipo_2 - tipo_2; (e) tipo_2 - tipo_3; (f) tipo_3 - tipo_3.*

Comportamento da velocidade relativa tangencial

Quando um corpo se movimenta em relação a outro, gerando contato com atrito, é lógico que sua velocidade relativa tangencial diminua constantemente até se anular. Nesta última fase, com velocidade relativa tangencial zero, os corpos estão em repouso e em contato. A velocidade relativa normal também pode ser zero, mas no paradigma de microcolisões, existe uma velocidade relativa normal mínima diferente de zero.

A Figura 7.6 mostra a evolução da velocidade da bola rolando sobre um bloco horizontal, apresentado pela Figura 7.4. O comportamento de atrito na dinâmica de contato para cada cena mostrada é do tipo (a) e (e) da Figura 7.5, respectivamente. Como é lógico, a bola rolando sobre o plano horizontal decrementa lentamente sua velocidade, apesar dos atritos constantes e variados, o que não ocorre com corpos deslizantes.

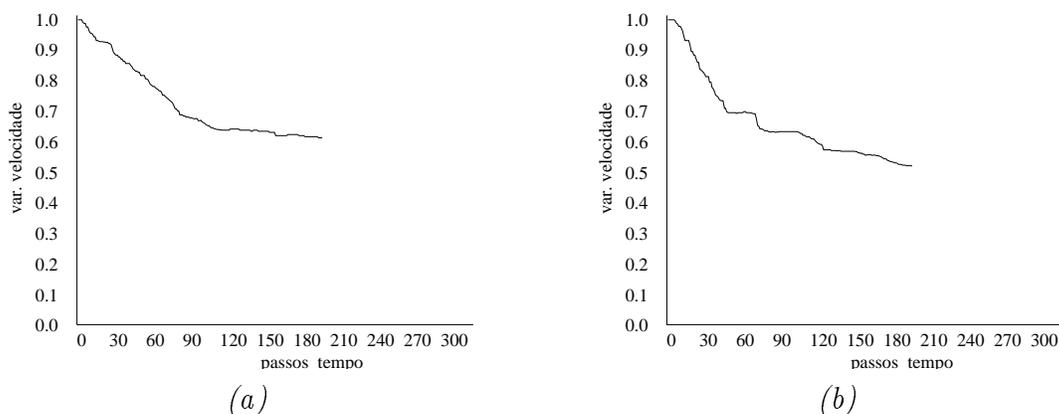


Figura 7.6: *Evolução da velocidade da bola rolando sobre uma superfície horizontal: (a) velocidade com atrito mostrado na Figura 7.5(a); (b) velocidade com atrito mostrado na Figura 7.5(e)*

A Figura 7.7 mostra um exemplo onde um objeto de geometria arbitrária é jogado sobre uma superfície horizontal estática, sobre a qual ele desliza até parar. No teste, o objeto superior tem uma velocidade tangencial relativa inicial de 10.00 unidades, que para efeitos do gráfico está normalizada para o intervalo unitário (10 a 1), de modo que esta velocidade diminui, dependendo das rugosidades, até atingir zero, tal como mostra a Figura 7.8. A curva na Figura 7.8(a) mostra o comportamento da velocidade relativa tangencial dos blocos em contato com rugosidades variadas do mesmo

tipo; a Figura 7.8(b) indica que a velocidade do corpo superior varia rapidamente na parte mais rugosa do bloco horizontal, e a velocidade é quase constante quando existe contato na parte de blocos com rugosidade quase nula. Neste último exemplo, considerou-se que o bloco superior tem rugosidade mínima e o bloco horizontal tem partes mais rugosas e partes lisas.

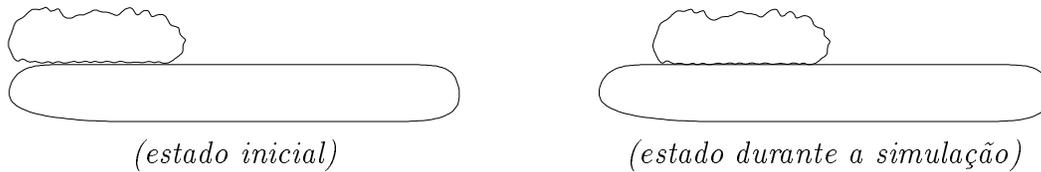


Figura 7.7: Um corpo de geometria irregular sendo deslizado sobre uma superfície horizontal estática.

No exemplo, observa-se que em cada caso de contato existem vários pontos de contato, que são tratados sequencialmente, e para cada ponto possivelmente existe um coeficiente de fricção diferente dos outros.

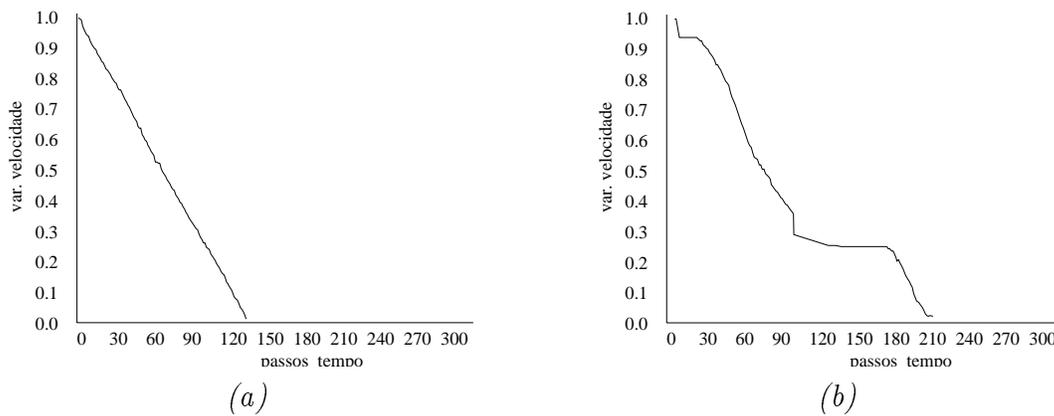


Figura 7.8: Comportamento da velocidade relativa tangencial de contato dos corpos da Figura 7.7: (a) quando ambos os corpos têm rugosidades variadas; (b) quando o corpo móvel tem rugosidade nula e o horizontal fixo tem rugosidade variada (rugosa, lisa e rugosa).

Comportamento de múltiplos corpos irregulares

Para exibir alguns exemplos de múltiplos objetos com superfícies irregulares, considerou-se, primeiramente, objetos com rugosidades das bibliotecas atribuídas nas suas superfícies refinadas; depois, apresentam-se outros exemplos com análise da dinâmica de contatos em objetos irregulares de múltiplas resoluções.

No primeiro caso, apresenta-se um exemplo de simulação do comportamento de vários corpos irregulares, composto de quatro corpos em movimento e dois corpos irregulares fixos. Foram atribuídos na suas superfícies refinadas rugosidades de *tipo_2* da biblioteca, e diferentes coeficientes de restituição ao contato para cada corpo da cena; com isto, o comportamento dos objetos corresponde ao esperado. A Figura 7.9 mostra a seqüência de comportamentos a cada 30 quadros de animação.

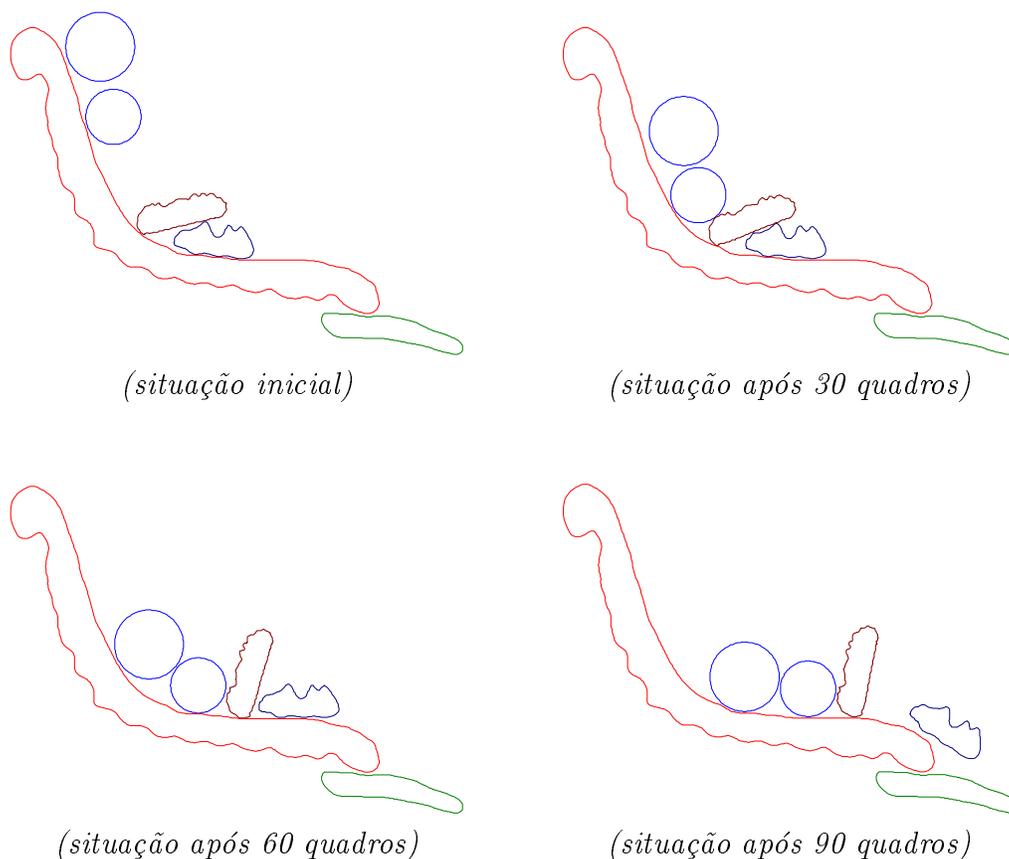


Figura 7.9: Simulação do comportamento de quatro objetos em movimento sobre um bloco fixo de superfície irregular. Cada corpo tem rugosidades variadas nas suas superfícies refinadas e diferentes coeficientes de restituição ao contato.

Um segundo exemplo de comportamento de múltiplos objetos irregulares, com rugosidades variadas (*tipo_2*) atribuídas às suas superfícies refinadas e diferentes coeficientes de restituição, é mostrado na Figura 7.10. A seqüência mostra um bloco horizontal fixo de superfície irregular com objetos livres, formando a palavra “gráfica”, e uma esfera se movimentando com uma velocidade inicial que colide com o objeto “g” da seqüência, distribuindo-os por efeitos de colisões múltiplas.

A Figura 7.11 mostra uma seqüência de situações nas que os objetos se encontram quando eles têm rugosidades variadas *tipo_3*. Observa-se que o comportamento dos objetos é um pouco diferente daquele da Figura 7.10 devido ao tipo de rugosidade de suas superfícies refinadas.

A Figura 7.12 mostra um conjunto de objetos de geometrias arbitrárias, representados em sua versão mais refinada e nas duas versões mais grosseiras de níveis vizinhas. O resultado das animações relativas a estes dois níveis grosseiras é mostrado na Figura 7.13. Observa-se que, conforme esperado, o nível de rugosidade (e, portanto, de atrito) é maior na versão de menor resolução, o que faz com que se observe uma pequena variação nos quadros relativos a esta versão.

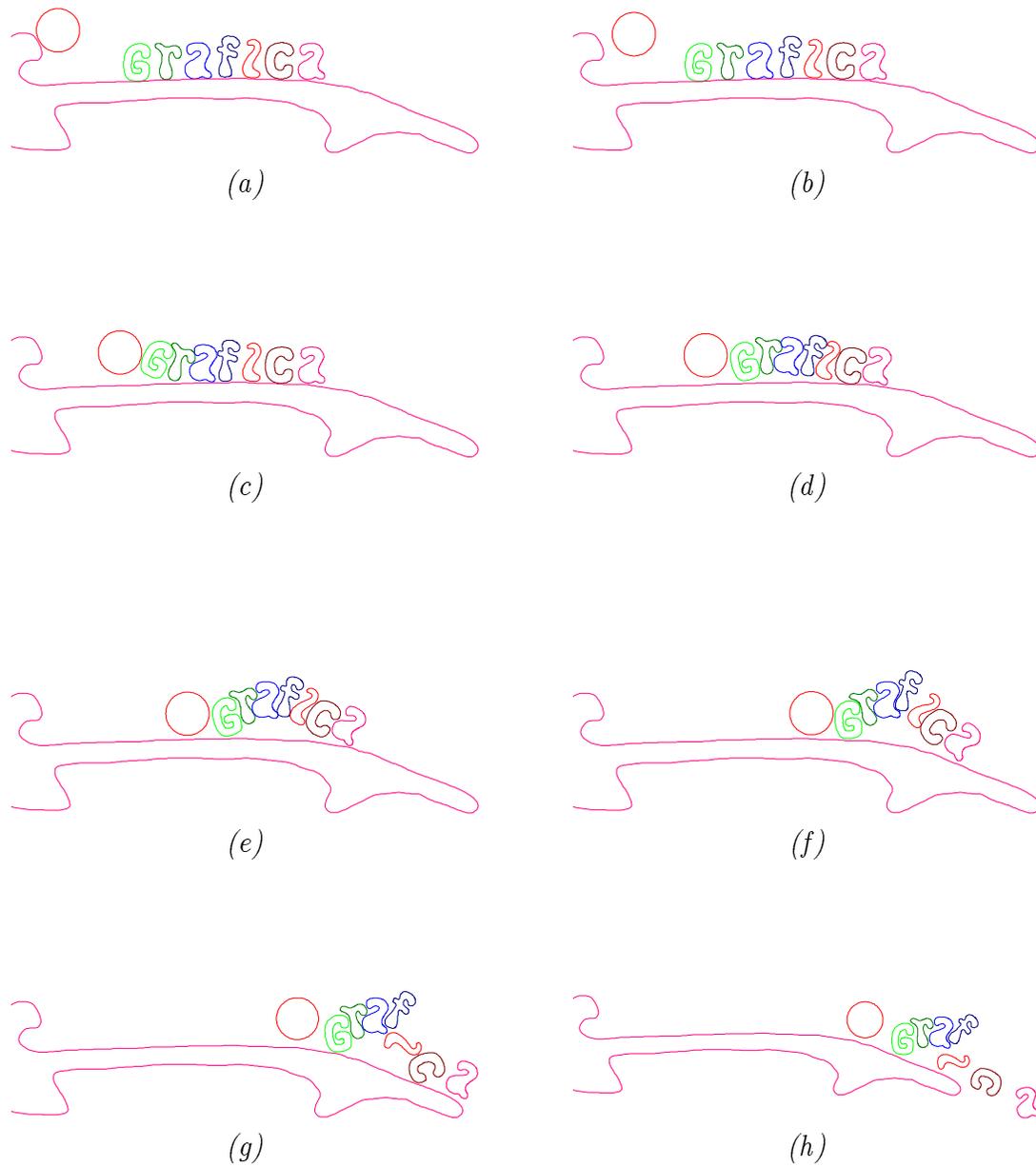


Figura 7.10: Simulação do comportamento de múltiplos objetos de superfícies refinadas irregulares com rugosidades tipo_2.

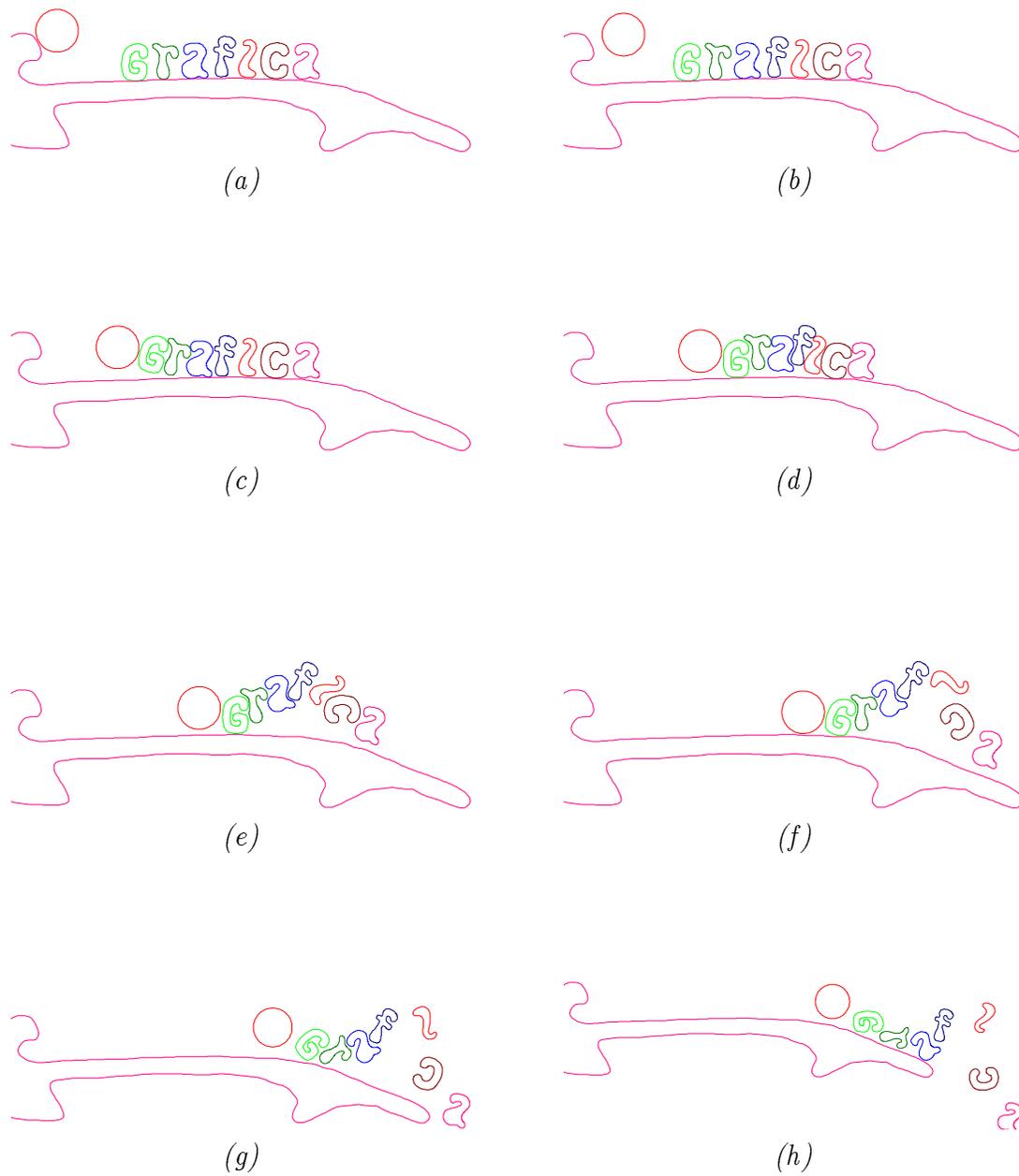
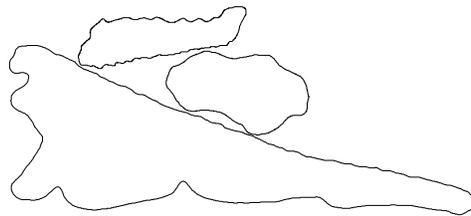
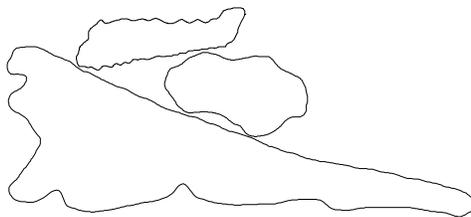


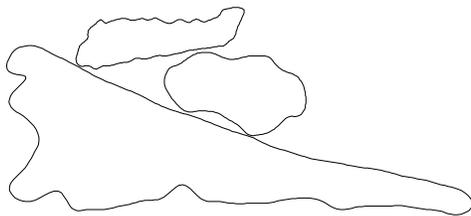
Figura 7.11: Simulação do comportamento de múltiplos objetos de superfícies refinadas irregulares com rugosidades tipo_3.



(a): versão refinada



(b): versão grosseira de primeiro nível



(c): versão grosseira de segundo nível

Figura 7.12: Um exemplo de três objetos de geometrias arbitrárias em três resoluções.

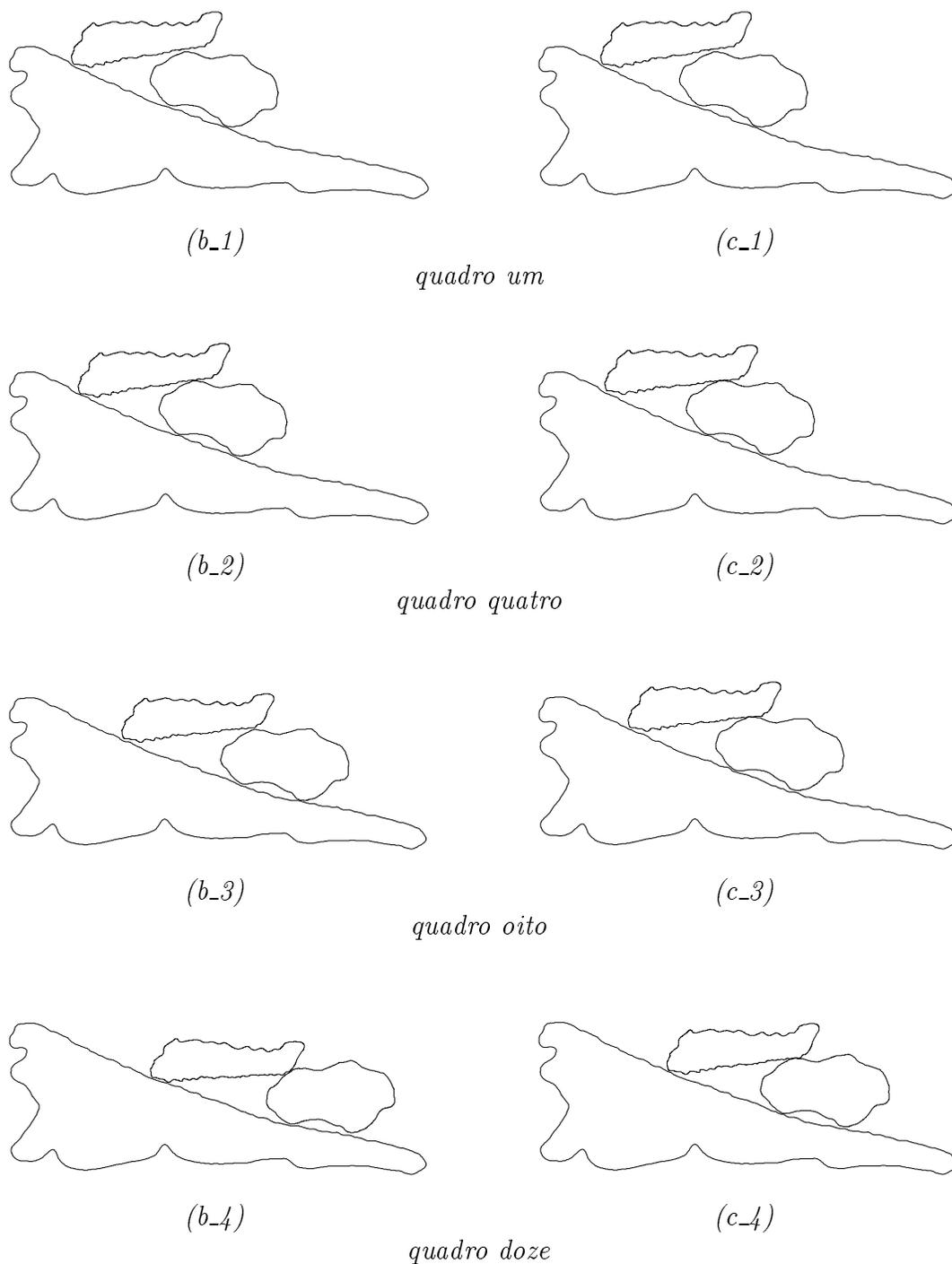


Figura 7.13: Seqüência de quadros gerados pela animação de três objetos rugosos definidos por sua geometria de primeiro e segundo nível de resolução: coluna b definida por resolução de primeiro nível (-1); coluna c definida por resolução de segundo nível (-2).

Capítulo 8

Conclusões e trabalhos futuros

Cada etapa do modelo abordado neste trabalho foi implementada na linguagem C, usando-se os pacotes gráficos IUP/LED e CD desenvolvidos pelo grupo de tecnologia de computação gráfica da PUC-Rio. Em cada capítulo se discutem mais de uma abordagem correspondente à etapa de referência, sendo que, em alguns casos estas são validadas com resultados comparativos após a sua respectiva implementação.

No projeto, não se incorporou a síntese da imagem realista, mas se formula um método para criar animações considerando os elementos básicos das características físicas da superfície dos objetos do mundo real, simulados em 2D. Neste capítulo, comentam-se as conclusões gerais e parciais do modelo proposto, e, finalmente, propõem-se trabalhos futuros para modelos de animação, e extensões dos modelos parciais para suas aplicações em outros problemas.

8.1 Conclusões

O trabalho desenvolvido propõe uma abordagem para modelar animações de objetos de geometria arbitrária, integrando as suas características geométricas. Com este objetivo, usam-se técnicas como análise de multiresolução dos objetos, que permite a criação eficiente de estruturas hierárquicas de caixas envolventes de segmentos de superfícies. Tais técnicas também são usadas com o objetivo de relacionar facilmente os coeficientes de detalhe com as rugosidades das superfícies dos objetos.

Os resultados dos exemplos mostrados validam a eficiência do modelo desenvolvido. Observa-se que a integração das características geométricas dos corpos com a dinâmica é fundamental para a produção de comportamentos interessantes dos corpos em animação. Uma pequena variação de alguma característica da superfície, como uma pequena perturbação, pode permitir a variação de seus comportamentos.

Uma combinação adequada de estruturas hierárquicas (estáticas) com procedimentos analíticos, na detecção de colisões, permite uma operação computacionalmente eficiente, em particular quando se envolvem objetos de geometria arbitrária.

Sendo de que o modelo desenvolvido é composto de vários sub-módulos, não se pode ignorar as contribuições de cada um desses sub-módulos. Note-se que as contribuições parciais, ao criarem diferentes modelos para a resolução de problemas parciais, auxiliaram no atingimento do objetivo final.

No Capítulo 2 abordou-se a definição de objetos de geometria arbitrária, tendo em vista que os objetos no mundo real se apresentam dessa forma. Objetivando-se a simulação dinâmica deste tipo de corpos, optou-se pela modelagem dos objetos em curvas B-splines cúbicas periódicas. A fim de se manipular a forma de objetos, formula-se um método de livre deformação de segmentos de curvas, criando-se uma forma de se movimentar minimamente os pontos de controle relativos ao ponto deslocado. Esta deformação local se estende para deformação global, baseada na distribuição da deformação nas resoluções inferiores do objeto representado em multi-resolução por transformação wavelet. Assim, um objeto de geometria arbitrária pode ser redefinido de diferentes formas para que o mesmo se aproxime de uma descrição do objeto do mundo real. Desta forma, evita-se o uso de objetos aproximados com faces poligonais ou por composição de objetos de geometria conhecida. A forma de redefinir a característica de um objeto criado por manipulação direta demonstra que os objetos do universo de simulação podem ser arbitrários, sem limitações de forma.

A mínima caixa retangular orientada, envolvendo todo o objeto, permite uma rápida projeção para os eixos coordenados do universo da simulação. Essas projeções permitem uma fácil comparação com outras caixas envolventes de outros objetos, de forma a se detectar objetos próximos. Este processo é um dos artifícios fundamentais na minimização de tempo na detecção de colisões entre dois objetos, já que permite ignorar aqueles objetos que não estão na situação de colidir.

A comparação pela interseção de um par de caixas orientadas, em uma estrutura de árvore binária, permite isolar rapidamente os pedaços de superfície que possivelmente estão em contato tolerado. Para isto, a definição de tolerância, para cada segmento mínimo da superfície, possui um papel muito importante. Já a sua verificação e os cálculos de pontos de contatos são feitos por procedimentos locais, usando-se métodos de subdivisão para a convergência rápida de métodos numéricos.

O MOBBtree para superfícies curvas, proposto neste trabalho, permite que os objetos de geometria arbitrária possam ser tratados como tais, sem serem aproximados a agregados de geometria conhecida ou segmentos poligonais. A forma de sua construção facilita a análise de interferência espacial em qualquer nível de resolução desejado dos objetos, sendo mais eficiente que a OBBtree devido a que as suas caixas orientadas são adaptadas ao comportamento dos pedaços das superfícies envolvidas.

A inclusão de tolerâncias nas caixas envolventes permite a identificação da rugosidade do segmento da superfície contida na caixa. Com isto, a caixa envolvente conserva as propriedades geométricas de cada pedaço da superfície do objeto, para que na análise de contatos sejam consideradas como elementos geradores dos atritos de contato. As propriedades conservativas das caixas básicas são generalizadas para as caixas superiores, em uma estrutura hierárquica de caixas, envolvendo um conjunto de segmentos básicos do objeto. A representação multiresolução do objeto facilita a criação dessas caixas superiores de forma que essas se adaptem e se ajustem aos respectivos segmentos da superfície.

O refinamento por subdivisão de uma superfície refinada requer o dobro do espaço usado por essa superfície. Neste sentido, a representação das rugosidades de níveis mais superiores são consideradas como ocorrendo em um intervalo de tolerância. Essas tolerâncias incluem as perturbações do segmento da superfície, que podem ser estimadas por procedimentos estatísticos. Em geral, neste trabalho se consideram as perturbações mais refinadas representando as rugosidades da superfície em referência, e estas, em cada segmento, definem as tolerâncias respectivas, as quais são indispensáveis na análise de contatos do segmento com outro similar.

A distribuição variada das tolerâncias na superfície do objeto permite a aproximação do comportamento das rugosidades locais. Esta aproximação facilita o cálculo de coeficientes de atritos variados na análise de contatos. Este atributo permite que o comportamento dinâmico dos objetos se aproxime da realidade daqueles fornecidos

pelas abordagens atuais, em que os coeficientes de atritos são constantes.

No tratamento de contatos baseado na conservação de momentos, não é necessário resolver um sistema de equações lineares, que demanda maiores recursos computacionais, sendo um dos fatores degradantes da performance da animação. O sistema de equações é reduzido a uma simples operação de avaliação de uma expressão de constantes. Este fato permite que, na animação, não se demande muito tempo nem espaço adicional, na resolução de sistemas lineares de equações pertinentes.

8.2 Trabalhos futuros

O modelo desenvolvido neste trabalho está orientado ao espaço tridimensional, à exceção da modelagem de objetos. Somente os exemplos, com o propósito de validar o trabalho, foram feitos em duas dimensões. Uma extensão imediata deste modelo para tridimensional, se concentraria na parte de criação dos objetos tridimensionais e sua representação em multiresolução.

A modelagem de objetos, elemento fundamental para que o sistema funcione em 3D, pode se estender para este caso, usando-se superfícies B-splines cúbicas. A parte da representação de objetos 3D em multiresolução é abordada em diferentes trabalhos referenciados no Capítulo 2. A manipulação direta dos objetos em 3D, com as técnicas formuladas neste trabalho, é uma das propostas de trabalhos futuros para o autor.

A representação de multiresolução do objeto pode ser feita com outras abordagens existentes, ao invés das transformadas wavelets, desde que essas mantenham a representação das médias de uma resolução para outra, condição necessária para a criação das caixas envolventes, conservando as características geométricas das superfícies, em hierarquia.

Também seria interessante a pesquisa a respeito da restrição do número de elementos (pontos de controle ou segmentos) em cada nível de resolução. Neste trabalho usaram-se elementos de potência de dois; seria, talvez, mais eficiente generalizar o modelo para que seja independente do número de elementos. Com este critério, na criação e manipulação da árvore de caixas orientadas, nem sempre se manipulariam árvores binárias balanceadas.

O exemplo da Figura 2.8 inspira a se aplicar o modelo formulado aqui no problema de “cortes” de peças nas manufaturas, de forma que cada objeto represente uma peça que se deseja cortar de um material inteiro, que pode ser couro para sapatos, tecidos, lâminas metálicas, cartões, etc. O objetivo dessas manufaturas é maximizar o número de peças de cortes, e minimizar os pedaços não aproveitáveis do material. A abordagem de Milenkovic [Milenko93] pode ser estendida usando-se o modelo proposto neste trabalho e adicionando-se modelos heurísticos, de forma que permita a movimentação de objetos até que todos fiquem em uma situação de contato contínuo (compacto). A heurística permitiria mover-se um objeto que esteja em uma determinada posição, impedindo a acomodação dos outros que poderiam se encaixar bem em outros lugares do universo da simulação. Todo o processo de movimentação dos objetos no trabalho de Milenkovic é baseado em simulação dinâmica (com colisões e contatos) com objetos poligonais. A tolerância também seria importante neste esquema, já que ela representaria a espessura do elemento de corte (faca), sendo que, nas partes de maior variação da derivada, a mesma deve ter valores maiores, devido à movimentação e dificuldades de giro do elemento de corte.

Animações com colisões de corpos, causando deformação quase física, também podem ser estendidas a partir deste modelo. Desde que toda a superfície do objeto esteja coberta por caixas hierárquicas, uma deformação de um segmento afetando um número de segmentos vizinhos, como visto no capítulo 2, permitiria recalcular rapidamente suas respectivas caixas, sem a necessidade de se recalcular todas as caixas da estrutura da árvore. A deformação dos segmentos vizinhos, como consequência da colisão, pode ser feita segundo a técnica de manipulação direta de pontos em contato. Assim, um ponto de contato se desloca para o interior do objeto em um segmento Δl , de forma que este seja originado do alongamento de uma suposta mola inserida no ponto de contato durante a ocorrência da colisão. Tendo um vetor de deslocamento Δl , e um ponto de contato sobre o corpo, pode-se aplicar a livre deformação formulada no Capítulo 2 deste trabalho, para simular uma deformação do corpo como consequência da colisão.

Outra extensão deste modelo poderia ser a de permitir a ocorrência de microdeformações nas superfícies dos objetos em fricção. Uma vez que os vetores de detalhe definem as perturbações sobre a superfície do objeto, eles podem ser modificados em magnitude e orientação. Essas modificações, baseadas nas forças de contato e direção de velocidades relativas tangenciais de contato, poderiam ser aplicadas sobre

os vetores de detalhe dos segmentos diretamente afetados pelo contato. Uma vez alterados os vetores de detalhe, após a reconstrução (transformada inversa wavelet), a superfície apresenta rugosidades alteradas. Este principio é interessante nas simulações de corpos rígidos, já que a superfície de um corpo, por mais rígida que seja, sempre experimenta microdeformações de seus atributos.

Apêndice A

Tabela de filtros

As semicolunas das matrizes de filtro, para multiresolução dos objetos, obtida pela análise de multiresolução biortogonal para B-splines cúbicas, se apresentam nesta seção. A relação de primal e dual denota-se por um par (*primal*, *dual*).

Para primal 4 (B-splines de ordem 4),

$$Pq = \begin{bmatrix} 0.375 \\ 0.25 \\ 0.0625 \end{bmatrix} \quad \text{e} \quad Bb = \begin{bmatrix} 0.375 \\ -0.25 \\ 0.0625 \end{bmatrix}.$$

Para dual 2,

$$Qq = \begin{bmatrix} 1.25 \\ -0.15625 \\ -0.375 \\ -0.09375 \end{bmatrix} \quad \text{e} \quad Aa = \begin{bmatrix} 1.25 \\ 0.15625 \\ -0.375 \\ 0.09375 \end{bmatrix}.$$

Para dual 4,

$$Qq = \begin{bmatrix} 1.09375 \\ -0.273437 \\ -0.375 \\ 0.00390625 \\ 0.078125 \\ 0.0195312 \end{bmatrix} \quad \text{e} \quad Aa = \begin{bmatrix} 1.09375 \\ 0.273437 \\ -0.375 \\ -0.00390625 \\ 0.078125 \\ -0.0195312 \end{bmatrix}.$$

Para dual 6,

$$Qq = \begin{bmatrix} 1.02539 \\ -0.320435 \\ -0.35791 \\ 0.0679932 \\ 0.112305 \\ 0.00671387 \\ -0.0170898 \\ -0.00427246 \end{bmatrix} \quad e \quad Aa = \begin{bmatrix} 1.02539 \\ 0.320435 \\ -0.35791 \\ -0.0679932 \\ 0.112305 \\ -0.00671387 \\ -0.0170898 \\ 0.00427246 \end{bmatrix} .$$

Para dual 8,

$$Qq = \begin{bmatrix} 0.986938 \\ -0.345428 \\ -0.342529 \\ 0.110291 \\ 0.127686 \\ -0.0125122 \\ -0.0324707 \\ -0.00331116 \\ 0.00384521 \\ 0.000961304 \end{bmatrix} \quad e \quad Aa = \begin{bmatrix} 0.986938 \\ 0.345428 \\ -0.342529 \\ -0.110291 \\ 0.127686 \\ 0.0125122 \\ -0.0324707 \\ 0.00331116 \\ 0.00384521 \\ -0.000961304 \end{bmatrix} .$$

Para dual 12,

$$Qq = \begin{bmatrix} 0.945082 \\ -0.371282 \\ -0.320374 \\ 0.161443 \\ 0.137804 \\ -0.0478584 \\ -0.0521088 \\ 0.00775886 \\ 0.014451 \\ 0.000261068 \\ -0.0025177 \\ -0.000373721 \\ 0.000204563 \\ 0.0000511408 \end{bmatrix} \quad \text{e} \quad Aa = \begin{bmatrix} 0.945082 \\ 0.371282 \\ -0.320374 \\ -0.161443 \\ 0.137804 \\ 0.0478584 \\ -0.0521088 \\ -0.00775886 \\ 0.014451 \\ -0.000261068 \\ -0.0025177 \\ 0.000373721 \\ 0.000204563 \\ -0.0000511408 \end{bmatrix} .$$

Nos processos de testes, na implementação, foram usados filtros (4, 8).

Bibliografia

- [AbraBaSa99] F. Abramovich, T. C. Bailey and T. Sapatinas. Wavelet Analysis and its Statistical Applications. *Technical Report*, Institute of Mathematics and Statistics, University of Kent at Canterbury, United Kingdom, <http://stork.ukc.ac.uk/IMS/statistics/people/T.Sapatinas/links/statistician.ps.gz>, 1999.
- [AbraBenj96] F. Abramovich and Y. Benjamini. Adaptive Thresholding of Wavelet Coefficients. *Computational Statistics & Data Analysis*, 1996, 22, 351-361.
- [AbraSilv98] F. Abramovich and B. W. Silverman. Wavelet Decomposition Approaches to Statistical Inverse Problems. *Biometrika*, 85, 1, 1998, 115-129.
- [Amirou92] F. M. Amirouche. Computational methods in multibody dynamics. *Prentice Hall, Englewood Cliffs, New Jersey*, 1992, 478 ps.
- [ArvoKirk89] J. Arvo and D. Kirk. A Survey of Ray Tracing Acceleration Techniques. *Introduction to Ray Tracing*, 1989, 201-262.
- [Balaban95] I. J. Balaban. An optimal algorithm for finding segments intersections. *Proceeding of the 11th Annual ACM Symposium on Computational Geometry*, 1995, 211-219.
- [Baraff89] David Baraff. Analytical methods for dynamic simulation of non-penetrating rigid bodies. *Computer Graphics*, v.23, n.3, 1989, 223-232.
- [Baraff90] David Baraff. Curved Surfaces and Coherence for Non-penetrating Rigid Body Simulation. *Computer Graphics*, v.24, n.4, 1990, 19-28.

- [Baraff92] David Baraff. Dynamic Simulation of Non-penetrating Rigid Bodies. *PhD. Thesis*, Department of Computer Science, Cornell University, Ithaca, 1992.
- [Baraff94] David Baraff. Rigid body simulation. *SIGGRAPH'94*, note of course, cd-ROM version, 1994, G1-G67.
- [Mirtich96] Brian Mirtich. Impulse-Based Simulation of Rigid Body System. *PhD. Thesis*, Department of Computer Science, University of California, Berkeley, 1996.
- [BarBea89] R. Bartels e J. Beatty. A Technique for the Direct Manipulation of Spline Curves. *Proceeding of Graphics Interface*, pp. 33-39, 1989.
- [BarFow93] R. Bartels e B. Fowler. Constraint Based Curve Manipulation. *IEEE Computer Graphics and Applications*, pp. 43-49, sept. 1993.
- [Celes98] Waldemar Celes. Efficient Assynchronous Evolution of Physical Simulations. *Proceeding of International Symposium on Computer Graphics, Image Processing and Vision, SIBGRAPI'98*, Rio de Janeiro, Brasil, 1998, 224-231.
- [ChaCan85] Steven C. Chapra and Raymond P. Canale. Numerical methods for engineers: with personal computer applications. *McGraw-Hill International Editions*, 1 ed., 1985, 570 ps.
- [Christo94] W. A. Christopher. Interactive multiple representation editing of physically-based animation. *PhD. Thesis*, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 1994.
- [CoqJan91] S. Coquillart e P. Jancène. Animated Free-Form Deformation: an interactive animation technique. *Computer Graphics Proceeding, SIGGRAPH'91*, pp. 23-26, 1991.
- [Cremer89] J. F. Cremer. An architecture for general purpose physical system simulation integrating geometry, dynamics and control, *PhD. Thesis*, Department of Computer Science, Cornell University, may, 1989.

- [deBerg+97] M. de Berg, M. van Kreveld, M. Overmars and O. Schwarzkopf. Computational Geometry: Algorithms and Applications. *Springer-Verlag*, 1985, 365 ps.
- [DoJoKePi95] D. L. Donoho, I. M. Johnstone, G. Kerkyacharian and D. Picard. Wavelet Shrinkage: Asymptopia?. *J. Royal Statistical Society B*, 57, 2, 1995, 301-369.
- [DoJoKePi96] D. L. Donoho, I. M. Johnstone, G. Kerkyacharian and D. Picard. Density Estimation by Wavelet Thresholding. *The Annals of Statistics*, 1996, Vol. 22, No. 2, 508-539.
- [DonoJohn94] D. L. Donoho and I. M. Johnstone. Ideal Spatial Adaptation by Wavelet Shrinkage. *Biometrika* 81, 3, 1994, 425-455.
- [DonoJohn95] D. L. Donoho and I. M. Johnstone. Adapting to Unknown Smoothness via Wavelet Shrinkage. *Journal of the American Statistical Association*, 1995, Vol. 90, No. 432, 1200-1224.
- [DonoJohn98] D. L. Donoho and I. M. Johnstone. Minimax Estimation via Wavelet Shrinkage. *The Annals of Statistics*, Vol. 26, No. 3, 1998.
- [Farin90] G. Farin. Curves and Surfaces for Computer Aided Geometric Design: A practical guide. *Academic Press, inc., San Diego*, 2da ed., 1990, 444 pgs.
- [FerPer96] G. Fernández and S. Periaswamy, LIFPACK: A software package for wavelet transforms using Lifting, *Technical Report TR-96-04*, Department of Computer Science, Univ. of South Carolina, Columbia, USA, 1996.
- [FiguStol97] L. H. de Figueiredo e J. Stolfi. Métodos numéricos autovalidados e aplicações. *21o Colóquio Brasileiro de Matemática*, julho 1997.
- [FinSal94] A. Finkelstein e D. Salesin. Multiresolution Curves. *Computer Graphics Proceedings*, SIGGRAPH 1994.
- [ForBar88] D. Forsey e R. Bartels. Hierarchical B-spline refinement. *Computer Graphics*, 22(4):205-212, 1988.

- [FucBis92] H. Fuchs e G. Bishop. Research Directions in Virtual Environments. *Computer Graphics*, 26(3):153-177, 1992.
- [FreeSha75] Freeman and R. Shapira. Determining the minimum area enclosing rectangle for an arbitrary closed curve. *ACM*, 18 (1975), pp. 409-413.
- [GleiKass92] M. Gleicher e M. Kass. An Interval refinement technique for surface intersection. *Proceeding of Graphics Interface*, pp. 242-249, 1992.
- [GomVel98] J. Gomes e L. Velho. From Fourier Analysis to Wavelets. *Course Notes (próximo)*, SIGGRAPH'98.
- [GoLiMa96] S. Gottschalk, M. Lin and D. Manocha. OBBtree: A Hierarchical Structure for Rapid Interference Detection. *Computer Graphics Proceeding, SIGGRAPH'96*, 1996, 171-180.
- [GuLiTa+97] Z. Guan, J. Ling, N. Tao, X. Ping e T. Ronghi. Study and Application of Physics-Based Deformable Curves and Surfaces. *Computer & Graphics*, v. 21, n. 3, pp. 305-313, 1997.
- [Hahn88] James Hahn. Realistic Animation of Rigid Bodies. *Computer Graphics*, 24(4), 1996, 299-308.
- [HallResn67] D. Halliday e R. Resnick. Physics. *John Wiley & Sons, inc., New York*, 2da ed., 1967, 1214 pgs.
- [Hoffmann89] C. Hoffmann. Geometric and Solid Modeling. *Morgan Kaufmann Publishers, Inc., San Mateo, CA*, 1989.
- [Hubbard95] Philip Hubbard. Collision Detection for Interactive Graphics Application. *PhD. Thesis*, Department of Computer Science, Brown University, Berkeley, 1995.
- [HuHuKa92] W. Hsu, J. Hughes e H. Kaufman. Direct manipulation of Free-Form Deformations. *Computer Graphics Proceeding, SIGGRAPH'92*, pp.177-184, 1992.
- [HsHuKa92] W. Hsu, J. Hughes e H. Kaufman. Direct Manipulation of Free-Form Deformations. *Computer Graphics Proceedings, SIGGRAPH 1992*.

- [JohnWich92] Richard A. Johnson and Dean W. Wichern. Applied Multivariate Statistical Analysis. *Prentice Hall*, 3 ed., 1992, 642 ps.
- [Kamat93] V.V. Kamat. A Survey of Techniques for Simulation of Dynamic Collision Detection and Response. *Computer & Graphics*, v.17, n.4, 1993, 379-385.
- [KazElb97] R. Kazinnik e G. Elber. Orthogonal Decomposition of Non-Uniform B-spline Space using Wavelets. *Eurographics'97*, pp. c26-c38, vol 16, num 3, 1997.
- [Leithol85] L. Leithol. Cálculo. *John Wiley & Sons, inc., New York*, 2da ed., 1985, 1214 pgs.
- [Lin94] Ming Lin. Efficient Collision Detection for Animation and Robotics. *PhD. Thesis*, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 1993.
- [Loop94] C. Loop. Smooth Spline Surfaces over Irregular Meshes. *Computer Graphics Proceeding*, 1994, 303-310.
- [Lotstedt81] Per Lötstedt. Coulomb friction in two dimensional rigid body systems. *Zeitschrift für Angewandte Mathematik und Mechanik*, 61(12), 1981, 605-615.
- [Lubachev91] B. Lubachevsky. How to Simulate Billiards and Similar Systems. *Journal of Computational Physics* 94, 1991, 255-283.
- [LycMor87] T. Lyche e K. Morken. Knot removal for parametric B-spline curves and surfaces. *Computer Aided Geometric Design*, 4(1987) 217-231.
- [MacJoy96] R. MacCracken e K. Joy. Free-Form Deformation with Lattices of Arbitrary Topology. *Computer Graphics Proceeding, SIGGRAPH'96*, pp. 181-188, 1996.
- [MacMilla36] W. D. MacMillan. Dynamics of Rigid Bodies. *Dover publications, inc., New York*, 1936, 478 pgs.
- [Milenko93] V. J. Milenkovic. Compactation and separation algorithms for non-convex polygons and their applications. *Technical Report*, Center for Research in Computing Technology, Harvard University, 1993.

- [Milenko96] V. J. Milenkovic. Position-Based Physics: Simulating the Motion of Many Highly Interacting Spheres and Polydra. *Computer Graphics Proceeding, SIGGRAPH'96*, pp. 181-188, 1996.
- [MirtCan94] Brian Mirtich e J. Canny. Impulse-Based Dynamic Simulation. *Tech. Report*, Department of Computer Science, University of California, Berkeley, 1994.
- [MoorWhil88] M. Moore and J. Wilhelms. An Collision Detection and Response for Computer Animation. *ACM Computer Graphics*, v.22, n.4, 1988, 289-298.
- [Morrison76] Donald F. Morrison. Multivariate statistical methods. *McGraw-Hill series in probability and statistics*, 2 ed., 1976, 415 ps.
- [Nason96] G. P. Nason. Wavelet Shrinkage using Cross-validation. *J. Royal Statistical Society B*, 1996, Vol. 58, No. 2, 463-479.
- [Piegl89] L. Piegl e N. Fog. Constrained B-spline curve and surface fitting. *Computer Aided design (CAD)*, v. 21, No. 10, pp. 641-648, 1989.
- [PieTil95] L. Piegl e W. Tiller. The NURBS Book. *Springer Editions, Berlin, New York*, 1995, 646 pgs.
- [PinhVida97] A. Pinheiro and B. Vidakovic. Estimating the Square Root of a Density via Compactly Supported Wavelets. *Computational Statistics & Data Analysis*, 25, 1997, 399-415.
- [PrepSham85] F. P. Preparata and M. L. Shamos. Computational Geometry, an Introduction. *Springer-Verlag*, 1985, 390 ps.
- [PrTeVe+92] William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery. Numerical Recipes in C: The Art of Scientific Computing. *Cambridge*, 2 ed., 1992, 994 ps.
- [RivCarVe99a] L. Rivera, P. C. Carvalho e L. Velho. Modelagem e manipulação de objetos de geometria complexa. *Relatório Técnico n. 6/99*, Departamento de Informática, Pontifícia Universidade Católica de Rio de Janeiro, 1999.

- [RivCarVe99b] L. Rivera, P. C. Carvalho e L. Velho. Interactive manipulation of multiresolution curves. IASTED International Conference in Computer Graphics and Imaging (CGIM'99), Palm Springs, CA, USA, october, 1999.
- [Rivera96] Luis 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.
- [RogAda90] D. Rogers e J. Adams. Mathematical Elements for Computer Graphics. *McGraw-Hill International Editions*, 2da ed., 1990, 611 pgs.
- [RogFog89] D. Rogers. Modifying the shape of rational B-splines. Part 1: curves. *Computer Aided design (CAD)*, v. 21, No. 8, pp. 509-518, 1989.
- [RubiWhil80] S. Rubin and T. Whitted. A 3-Dimensional Representation for Fast Rendering of Complex Scenes. *Computer Graphics Proceeding, SIGGRAPH'80*, 1980, 110-116.
- [SedPar86] T. Sederberg e S. Parry. Free-Form Deformation of Solid Geometric Models. *Computer Graphics Proceeding, SIGGRAPH'86*, pp. 151-160, 1986.
- [Snyder95] J. M. Snyder. An Interactive Tool for Placing Curved Surfaces without Interpenetration. *Computer Graphics Proceeding, SIGGRAPH'95*, 1995, 209-218.
- [StDeSa96] E. Stollnitz, T. Derose e D. Salesin. Wavelets for Computer Graphics: Theory and applications. *Morgan Kauffmann Publishers, inc.*, 1996, 245 pgs.
- [Sweldens96] W. Sweldens The Lifting scheme: a construction of second generation wavelets, *SIAM Journal on Mathematical Analysis*, nov, 1996.
- [SweSch95] W. Sweldens e P. Schroder. Building Your Own Wavelets at Home. *Technical Report 1995:5*, Department of Mathematics, Univ. of South Carolina, 1995.

- [SzelTerz89] R. Szeliski e D. Terzopoulos. From Splines to Fractals. *ACM Computer Graphics*, v.23, n.3, 1989, 51-60.
- [Taubin95] G. Taubin. A signal Processing Approach to Fair Surface Desing. *Computer Graphics Proceeding, SIGGRAPH'95*, 1995, 351-358.
- [TeneAuge86] A. M. Tenenbaum and M. J. Augenstein. Data Structures Using Pascal. *Prentice-Hall International Editions*, 2 ed., 1986, 774 ps.
- [TerWit88] D. Terzopaulos e A. Witkin. Physically based models with rigid and deformable components. *IEEE Computer Graphics and Application*, v.8, n.6, 1988, 41-51.
- [ThuWut97] G. Thurmer e A. Wuthrich. Normal Computation for Discrete Surfaces in 3D Space. *Eurographics'97*, vol 16, num 3, 1997.
- [Triboule95] K. Tribouley. Practical Estimation of Multivariate Densities Using Wavelet Methods. *Statistica Neerlandica*, 1995, Vol. 49, No. 1, 41-62.
- [Welch95] W. Welch, Serius Putty: Topological Design for Variational Curves and Surfaces, *PhD Thesis*, Department of Computer Science, Carnegie Mellon University, Pittsburgh, USA, 1995.
- [WelWit94] W. Welch e A. Witkin. Free-Form Shape Desing Using Triangulated Surfaces. *Computer Graphics Proceeding, SIGGRAPH'94*, pp. 247-256, 1994.
- [Wesseli96] W. Wesselink, Variational Modeling of Curves, *PhD Thesis*, Department of Computer Science, Eindhoven University of Technology, 1996.
- [Witkin94] A. Witkin. Particles system simulation. *SIGGRAPH'94*, note of course, cd-ROM version, 1994, A-C.
- [Wood85] D. Wood. An isothetic view of computational geometry. *Machine Intelligence and Pattern Recognition 2*, Computational Geometry, ed. G. Toussaint, 1985, 429-459.
- [WeHoGre84] H. Weghorst, G. Hooper and D. Greenberg. Improved Computational Methods for Ray Tracing. *ACM Transactions on Graphics*, 1984, 52-69.

- [ZorSchSw96] D. Zorin, P. Schröder and W. Sweldens. Interpolating Subdivision for Meshes whit Arbitrary Topology. *Computer Graphics Proceeding, SIGGRAPH'96*, 1996.
- [ZoSchSw97] D. Zorin, P. Schröder and W. Sweldens. Interactive Multiresolution Mesh Editing. *Computer Graphics Proceeding, SIGGRAPH'97*, 1997, 259-268.