

Universidade Federal de Campina Grande
Centro de Ciências e Tecnologia
Programa de Pós-Graduação em Matemática
Curso de Mestrado em Matemática

Modelos Deformáveis de Partículas e Algoritmos de Colisões Aplicados à Simulação de Tecidos

por

Jamilson Ramos Campos

sob orientação de

Profa. Dra. Rosana Marques da Silva
Prof. Dr. Vanio Fragoso de Melo

Dissertação apresentada ao Corpo Docente do Programa de Pós-Graduação em Matemática - CCT - UFCG, como requisito parcial para obtenção do título de Mestre em Matemática.

Campina Grande - PB

Dezembro/2006

Modelos Deformáveis de Partículas e Algoritmos de Colisões Aplicados à Simulação de Tecidos

por

Jamilson Ramos Campos

Dissertação apresentada ao Corpo Docente do Programa de Pós-Graduação em Matemática - CCT - UFCG, como requisito parcial para obtenção do título de Mestre em Matemática.

Área de Concentração: Matemática Aplicada

Aprovada por:

Prof. Dr. Adelailson Peixoto da Silva

Prof. Dr. Herman Martins Gomes

Profa. Dra. Rosana Marques da Silva
Orientador

Prof. Dr. Vanio Fragoso de Melo
Orientador

Universidade Federal de Campina Grande
Centro de Ciências e Tecnologia
Programa de Pós-Graduação em Matemática
Curso de Mestrado em Matemática

Dezembro/2006

Resumo

Este trabalho apresenta um estudo de dois modelos computacionais deformáveis aplicados à simulação de tecidos, ambos modelos de mecânica de partículas fisicamente embasados, contemplando algoritmos, um para cada modelo, para o tratamento de colisões. Estudamos um modelo de malha retangular (clássico e pouco robusto) e um baseado em malha triangular (moderno e robusto) através de implementação, simulações e uma análise qualitativa simples entre os resultados visuais obtidos com ambos. Nenhum destes modelos apresenta relações entre deformações tangenciais e normais e, portanto, não geram rugas e/ou dobras espontaneamente. Para torná-los mais realísticos, em nossa implementação, propomos o uso de uma força de acoplamento entre as deformações tangenciais e normais.

Abstract

This work presents a study of two deformable computational models applied to simulation of cloths, both physically based models of particle mechanics, contemplating, one per model, collision treatment algorithms. We study a rectangular grid model (classic and not very robust one) and a triangular mesh based model (modern and robust one) through implementation, simulations and a simple qualitative analysis between visual results reached with them. Neither of these models presents a relationship with tangent and normal deformations therefore don't build folds and/or wrinkles spontaneously. To turn then more realistic, in our implementation, we propose to make use of a coupling force between tangent and normal deformations.

Agradecimentos

À Deus, força criadora do universo.

À minha mãe, Edorice Ramos Campos, pela vida, amor, educação, paciência, conforto e oração. Sou-lhe muito grato por ter-me feito homem.

À minha esposa Andréa Dantas pela amor, paciência e compreensão ao longo de nossa vida a dois e no curso desse mestrado. À minha filha Raquel pelo carinho e divertimento.

Aos meus irmãos, tios e demais familiares pelo afeto, amizade, apoio, pelas comemorações e momentos de lazer que nos fazem relaxar e esquecer os problemas da vida.

Aos professores do DM/UFPB pela minha formação em João Pessoa. Em especial aos professores Marivaldo Matos e Antônio de Andrade e Silva que sempre me apoiaram nessa etapa da minha formação. Ao professor Abdoral de Souza pelas dicas, livros e apoio.

Aos meus orientadores Rosana Marques e Vanio Fragoso pelo conhecimento, pela acolhida, confiança e amizade. Fico feliz em ser vosso discípulo no “admirável mundo novo” da computação gráfica.

Aos professores e funcionários da Pós-graduação em Matemática da UFCG, que contribuíram direta ou indiretamente na minha formação e neste trabalho. Ao professor Daniel Pellegrino por sua gentil disponibilidade e apoio.

Aos professores Adelailson Peixoto e Herman Martins, membros da banca examinadora, pelo apoio e contribuições valorosas a este trabalho.

À professora Liliane Machado do DI/UFPB que, apesar de muito atarefada, me auxiliou no estudo do GL e C++.

À direção do CCEN/UFPB e a todos os meus colegas de trabalho pelo auxílio inestimável a esse meu propósito. Em especial ao amigo e colega Aleksandro Coutinho.

Aos amigos e companheiros de Campina e João Pessoa que deram-me força e

ânimo. Em especial à Moisés, Jonathas, Fábio Dantas, Clarissa, Romero, Fernanda, Marcos, Amanda, Suene e Júlio.

“Dizem que o desejo de conhecimento nos fez perder o paraíso no passado; verdade ou não, é certo que nos dará o paraíso no futuro.”

Ingersoll

Dedicatória

Aos meus pais Sebastião e Edorice, aos meus irmãos, à minha esposa Andréa e aos meus filhos Raquel e Tales.

,

Sumário

1	Introdução	1
1.1	Apresentação	1
1.2	Motivação	2
1.3	Classificação dos Modelos Deformáveis	2
1.4	Objetivos	3
1.5	Contribuições	3
1.6	Metodologia	3
1.7	Organização da Dissertação	4
2	Modelos Deformáveis de Partículas	5
2.1	Princípio e Representação	5
2.2	Revisão Bibliográfica	7
3	Modelo Baseado em Malha Retangular	11
3.1	O Modelo Massa-Mola	11
3.2	O Efeito “Super-Elástico”	14
3.3	Controle da Deformação por Dinâmica Inversa	16
3.4	Resultados e Discussões	18
4	Modelo Baseado em Malha Triangular	21
4.1	O Modelo Massa-Mola	21
4.1.1	Elementos de Curvatura	22
4.2	Integração Mista	27
4.3	Controle da Deformação	28
4.4	Controle da Taxa de Deformação	30

4.5	Resultados e Discussões	30
5	Algoritmo de Colisão Para o Modelo de Malha Retangular	35
5.1	Detecção de Colisões	35
5.1.1	A Colisão Ponto-Triângulo	36
5.1.2	A Colisão Aresta-Aresta	37
5.2	Otimizações da Detecção de Colisões	38
5.2.1	Hierarquia de Caixas Envolventes	38
5.2.2	A Curvatura da Superfície e a Detecção de Auto-colisões	40
5.3	Resposta à Colisão	41
5.3.1	Contato e Fricção	41
5.3.2	Impacto e Dissipação	42
5.3.3	Resposta Total	43
5.4	Consistência para Múltiplas Colisões	43
5.4.1	Zonas de Impacto	44
5.4.2	Hipótese de Colisão para Zonas de Impacto	45
5.4.3	Iteração	45
5.5	Resultados e Discussões	46
6	Algoritmo de Colisão Para o Modelo de Malha Triangular	51
6.1	Detecção de Proximidade e Colisão	51
6.2	Resposta ao Contato e à Colisão	53
6.2.1	Determinando Velocidades e a Aplicação de Impulsos	53
6.2.2	Repulsão	54
6.2.3	Fricção	56
6.2.4	Colisão	56
6.2.5	Zonas de Impacto	57
6.3	Aplicação do Algoritmo	58
6.4	Resultados e Discussões	59
7	A Força de Acoplamento	65
7.1	O Problema	65
7.2	Aplicação da Força de Acoplamento	66

8	Conclusões e Futuros Trabalhos	69
8.1	Conclusões	69
8.2	Considerações Finais	70
8.3	Futuros Trabalhos	71
A	Métodos Numéricos	73
B	Produto Tensorial e Tensores	75
C	Quadros de Animações	77
D	Detalhes Implementativos	81
	Bibliografia	83

Lista de Figuras

2.1	Treliça massa-mola representando um sistema de partículas.	6
2.2	Parâmetros mecânicos do modelo.	7
2.3	Comprimentos de molas ou ângulos versus deformações no sistema de partículas.	7
3.1	Os tipos de molas ligando $\mathbf{P}_{i,j}$ a seus vizinhos.	12
3.2	Deformação do modelo elástico: bandeira pendurada por dois cantos. . .	15
3.3	Controle da deformação por dinâmica inversa	17
3.4	Correção das molas super-elongadas.	18
3.5	Instabilidade no exemplo da Seção 3.2.	18
3.6	Energia indevida produzida pela correção da posição.	19
3.7	Modelo versus dobras espontâneas.	20
3.8	Mais exemplos: bandeira e vela ao vento.	20
4.1	Malha triangular.	22
4.2	Elemento de curvatura com ângulo diedral $\pi - \theta$	23
4.3	Modos de movimento para o elemento de curvatura.	23
4.4	Interpretação geométrica do fator $\frac{ \mathbf{E} ^2}{ \mathbf{N}_1 + \mathbf{N}_2 }$	25
4.5	Ângulos de repouso permitem pré-esculpir dobras nos tecidos.	26
4.6	Interpretação geométrica para a aproximação (4.3) de $d\theta/dt$	27
4.7	Controle da deformação: ajuste das velocidades com conservação do momento linear.	29
4.8	Deformação do modelo elástico: bandeira pendurada por dois cantos. . .	32

4.9	Bandeiras ao vento em modelos distintos: (a) malha retangular; (b) malha triangular.	33
4.10	O modelo não gera dobras espontaneamente: (a) pontos distribuídos no plano; (b) pontos com distribuição não planar.	33
4.11	Mais um exemplo para o modelo de malha triangular: vela ao vento. . .	34
5.1	Colisão ponto-triângulo: (a) antes da colisão; (b) durante a colisão. . .	37
5.2	Colisão aresta-aresta: (a) antes da colisão; (b) durante a colisão.	38
5.3	Divisão recursiva do tecido em zonas e armazenamento numa árvore binária.	39
5.4	Caixa alinhada com os eixos envolvendo uma primitiva.	39
5.5	Cone de normais para uma zona.	40
5.6	Cone de ângulo α calculado a partir dos cones descendentes de ângulos α_1 e α_2	41
5.7	Circunscrição iterativa de zonas de impacto.	44
5.8	Tecido caído sobre mesa quadrada (a) e retangular (b).	46
5.9	Colisão entre o tecido e uma mesa redonda.	47
5.10	Tecido caído sobre esferas de raios diferentes.	47
5.11	Tecido deslizando sobre um plano inclinado (a) e uma esfera (b).	48
5.12	O tecido fica “preso” à borda de um plano quando este é inclinado. . . .	49
6.1	A medida da sobreposição entre regiões do tecido: d representa a sobreposição e h a espessura do tecido.	56
6.2	Tecido caído sobre mesa quadrada (a) e retangular (b).	59
6.3	Colisão entre o tecido e uma mesa redonda.	59
6.4	Tecido deslizando sobre um plano inclinado (a) e uma esfera (b).	60
6.5	Colisão de arestas: (a) com o algoritmo de Provot e (b) com o algoritmo de Bridson et al.	61
6.6	Em (a), com o algoritmo [38], não há deslizamento e em (b), com o algoritmo [10], ocorre o deslizamento.	61
6.7	O tecido não fica preso à borda do plano.	62
6.8	Situações-problema para o algoritmo de Provot [38].	62
6.9	Dobras estáveis aumentam o realismo.	63

7.1	As dobras não aparecem espontaneamente.	65
7.2	Mais uma situação onde rugas não surgem.	66
7.3	Região a de aplicação da força de acoplamento.	67
7.4	Resposta coerente do modelo com forças de acoplamento.	68
C.1	Vela ao vento com o modelo de malha retangular.	77
C.2	Vela ao vento com o modelo de malha triangular.	78
C.3	Colisões para malha retangular: tecido colidindo com mesa quadrada. .	78
C.4	Colisões para malha triangular: tecido colidindo com mesa quadrada. .	79

Capítulo 1

Introdução

Neste capítulo apresentamos os objetos de nosso estudo, uma motivação, uma classificação e uma breve discussão a seu respeito. Ponderamos os objetivos e a metodologia do nosso trabalho e oferecemos um esquema de sua organização.

1.1 Apresentação

Este trabalho apresenta o estudo de dois modelos deformáveis baseados em mecânica de partículas para uso em simulações de tecidos: O modelo de Provot [37], de malha retangular, e o de Bridson et al. [11], de malha triangular. O primeiro modelo não oferece um tratamento para colisões (auto-colisões e com outros objetos) assim estudamos o algoritmo [38], do mesmo autor, desenvolvido posteriormente para esse modelo. O segundo modelo remete o tratamento de auto-colisões ao artigo [10] mas já traz consigo o tratamento de colisões com objetos. Por simplicidade, optamos, em nosso estudo, por implementar para esse modelo deformável o algoritmo de colisões [10].

A implementação desses modelos e algoritmos faz parte deste trabalho e o completa, possibilitando a submissão dos mesmos a uma variedade de testes e situações. Numa dessas situações, à propósito, observamos um fenômeno de falta de realismo nos dois modelos: eles não geram dobras e/ou rugas espontaneamente. Oportunamente descrevemos o problema e a solução proposta para tal.

1.2 Motivação

Há décadas a comunidade de computação gráfica estuda a modelagem de objetos deformáveis em aplicações diversas gerando imagens, animações e cenas virtuais comparáveis às cenas reais. Essas aplicações incluem a geração de imagens médicas, animação de músculos, simulações cirúrgicas para treinamento de pessoal e simulação de tecidos para vestimentas de atores virtuais [22].

A modelagem e animações de tecidos, nosso objeto de estudo, é de grande importância para as indústrias do entretenimento e da moda. O projeto de vestuários através da animação de tecido sobre figuras humanas, sobre outros objetos ou ainda suspensos são algumas aplicações desses modelos às indústrias da moda e têxtil. Para o entretenimento destacamos a animação de vestimentas sobre atores virtuais [26, 36, 46].

1.3 Classificação dos Modelos Deformáveis

Os modelos deformáveis podem ser classificados em três tipos: modelos geométricos [36, 4, 14], físicos [1, 34, 37, 11, 7, 3, 42] e híbridos [19, 15]. Os modelos geométricos usam transformações e entidades geométricas e não consideram as propriedades físicas do objeto. São úteis para modelagem de objetos rígidos estáticos. Os modelos físicos usufruem de conceitos e propriedades físicas para a dinâmica do objeto e, por isso, são realistas e mais intuitivos. Têm por desvantagem o alto custo computacional quando comparado aos modelos geométricos. Os modelos híbridos incorporam técnicas de modelagem geométrica para diminuir o tempo de processamento em modelos físicos, substituindo-os em momentos em que uma técnica geométrica aproxima bem o comportamento do modelo. Em nosso trabalho os objetos são fisicamente modelados.

Pela representação geométrica do objeto, na qual é desenvolvida a teoria da deformação, subdividimos os modelos físicos em dois ramos: os que consideram o objeto como um conjunto discreto, como no caso dos modelos de sistemas de partículas [37, 11] utilizando a mecânica de partículas, e os que consideram o objeto como um contínuo [42, 12, 33], utilizando a mecânica dos contínuos. Os do primeiro ramo partem do menor elemento constituinte do objeto, a partícula ou ponto-massa, interligados aos vizinhos por fios ou molas constituindo um mecanismo para modelar as interações entre elementos vizinhos. Os modelos do segundo ramo, baseados no contínuo, geralmente

representam o objeto parametricamente e consideram uma região contínua em torno de cada ponto do objeto onde são impostos princípios de equilíbrio e obtidas equações diferenciais parciais que governam esse equilíbrio. Nessa abordagem, os parâmetros de controle da deformação do objeto já vêm embutidos no modelo e não há uma busca de modelar as possíveis interações internas como ocorre nos modelos de partículas. Nesse sentido, os modelos baseados em mecânica dos contínuos são considerados *geometricamente exatos* e os de mecânica de partículas *geometricamente inexatos* [41]. Nossos objetos de estudo possuem representações geométricas discretas.

1.4 Objetivos

O principal objetivo deste estudo é adquirir entendimento, familiarização e assimilação dos conceitos, da problemática e das vantagens inerentes à modelagem de tecidos através dos modelos deformáveis de partículas e dos algoritmos de colisões implementados.

1.5 Contribuições

A principal contribuição deste trabalho para os modelos deformáveis de partículas é a força de acoplamento entre deformações tangenciais e normais (descrita no Capítulo 7). Essa força é usada como solução para o problema, detectado em ambos os modelos estudados, na geração de rugas e dobras espontâneas.

Também podemos citar como contribuições o desenvolvimento do softwares dedicados à modelos de sistemas de partículas e de apoio. Desenvolvemos bibliotecas para cada modelo e algoritmo de colisão além de bibliotecas de funções matemáticas adicionais, de estruturas de dados e de visualização.

1.6 Metodologia

Com o estudo e implementação de um modelo de partículas clássico, como o de Provot [37], temos um ponto de partida ideal para o nosso intuito. De fato, não se constrói um edifício pelo topo. Precisamos de uma base para estudos mais avançados, sendo esse primeiro modelo um ótimo referencial. Após esse passo, naturalmente se

procura um representante moderno dessa classe de modelos deformáveis. Nossa escolha recaiu então sobre o modelo de Bridson et al. [11], um modelo atual e robusto. Buscamos manifestações de falta de realismo em ambos e propomos modificações para a solução desses problemas. Estudamos, em seguida, algoritmos de colisões para ambos os modelos. Eles permitem a modelagem da interação do tecido com objetos em cena (inclusive o próprio tecido) sendo, assim, indispensáveis para praticamente todos os tipos de simulações.

De acordo com nossos objetivos, faremos, em nossos estudos, um comparativo qualitativo entre os conjuntos modelo-colisões. Isso não será feito do ponto de vista de eficiência nem do ponto de vista numérico. Apenas aspectos visuais e de realismo nas simulações serão observados. Discutiremos também algumas abordagens e questões inerentes à implementação, quando necessárias.

1.7 Organização da Dissertação

No Capítulo 2 apresentamos uma breve revisão bibliográfica dos modelos deformáveis de partículas e algoritmos de colisões focados em aplicações para simulação de tecidos. No Capítulo 3 apresentamos o modelo de malha retangular e no Capítulo 4 descrevemos o modelo de malha triangular. Os Capítulos 5 e 6 tratam dos algoritmos de colisões para o primeiro e segundo modelos, respectivamente. O Capítulo 7 apresenta a nossa proposta da força de acoplamento; solução para um problema de falta de realismo em ambos os modelos. O Capítulo 8 traz conclusões sobre o trabalho e propostas de futuros estudos relacionados à área. O Apêndice A apresenta uma discussão sobre métodos numéricos relacionados ao nosso trabalho, o apêndice B apresenta uma descrição sucinta de tensores e do produto tensorial de vetores, o apêndice C exhibe quadros da animação de alguns exemplos do trabalho bem como uma tabela com seus tempos de processamento e, finalmente, o apêndice D relata alguns detalhes implementativos.

Modelos Deformáveis de Partículas

Faremos aqui uma revisão bibliográfica de trabalhos em modelos de partículas fisicamente embasados e direcionados à simulação de superfícies deformáveis; em nosso caso, tecidos. Antes disso, expomos uma descrição do princípio e da representação desses modelos.

2.1 Princípio e Representação

Em vez de considerar as propriedades mecânicas do material como um todo, os modelos de sistemas de partículas discretizam o material em um conjunto de pontos-massa, que interagem através de forças, modelando aproximadamente o comportamento do mesmo.

Em aplicações de computação gráfica, superfícies são comumente representadas por malhas poligonais. Assim, uma maneira conveniente de construir um sistema de partículas é considerar cada vértice da malha como um ponto-massa e as arestas dos polígonos como forças de interação entre as diversas partículas. Cada partícula é representada por sua posição, velocidade, aceleração, massa e outros atributos.

A maneira usual de simulação numérica da dinâmica de deformação e movimento de um sistema de partículas é por integração da segunda lei de Newton [25]

$$\mathbf{F}(t) = M \frac{d^2 \mathbf{P}}{dt^2}$$

onde \mathbf{P} é a posição da partícula, \mathbf{F} o somatório das forças aplicadas à partícula e M a sua massa. Disso advém uma grande vantagem do uso de sistema de partículas: sua

simplicidade. As posições, velocidades e forças das partículas aparecem em equações diferenciais ordinárias de fácil resolução.

As forças aplicadas a cada partícula dependem do estado momentâneo do sistema, representado pelas posições e velocidades de cada partícula. Estas forças representam todos os efeitos mecânicos sobre o sistema, incluindo a gravidade, efeitos aerodinâmicos e quaisquer outras influências externas. Entretanto, alguns efeitos geométricos (como interações com outros objetos ou com ele próprio) podem influir diretamente na posição e velocidade de uma partícula, como por exemplo o tratamento de colisões.

Os sistemas massa-mola são representações mais comuns para os parâmetros mecânicos em um sistema de partículas (Figura 2.1). Nessa abordagem, cada partícula é ligada às vizinhas por molas que representam o comportamento elástico do material através da lei de Hook [25]. Essas molas tendem a manter as partículas em suas posições iniciais (ou de repouso). Vários tipos de molas podem ser usadas para representar diversos parâmetros mecânicos do modelo como as alterações métricas de esticamento e compressão, o cisalhamento e a curvatura ou arqueamento (Figura 2.2).

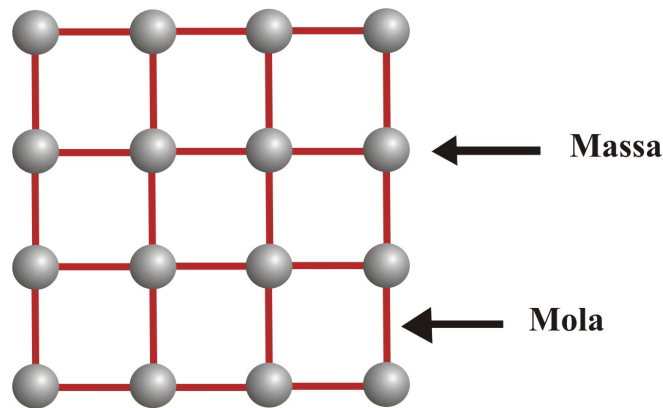


Figura 2.1: Treliça massa-mola representando um sistema de partículas.

Enquanto que a elasticidade métrica pode ser modelada por molas de alongação ao longo das arestas, a elasticidade de cisalhamento pode ser modelada tanto pelos ângulos entre as molas como por molas de alongação diagonais. Da mesma forma, a elasticidade de curvatura pode ser modelada por ângulos entre arestas opostas ou por molas de flexão entre vértices opostos (Figura 2.3).

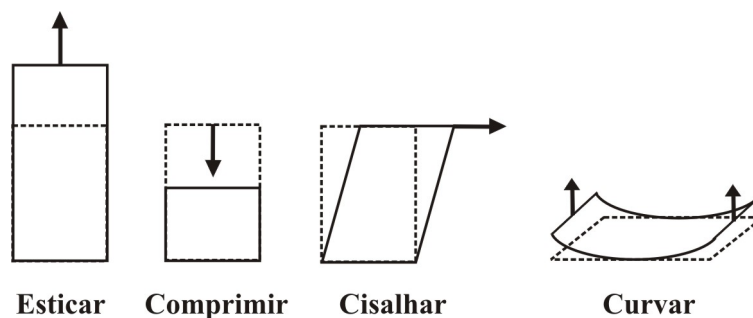


Figura 2.2: Parâmetros mecânicos do modelo.

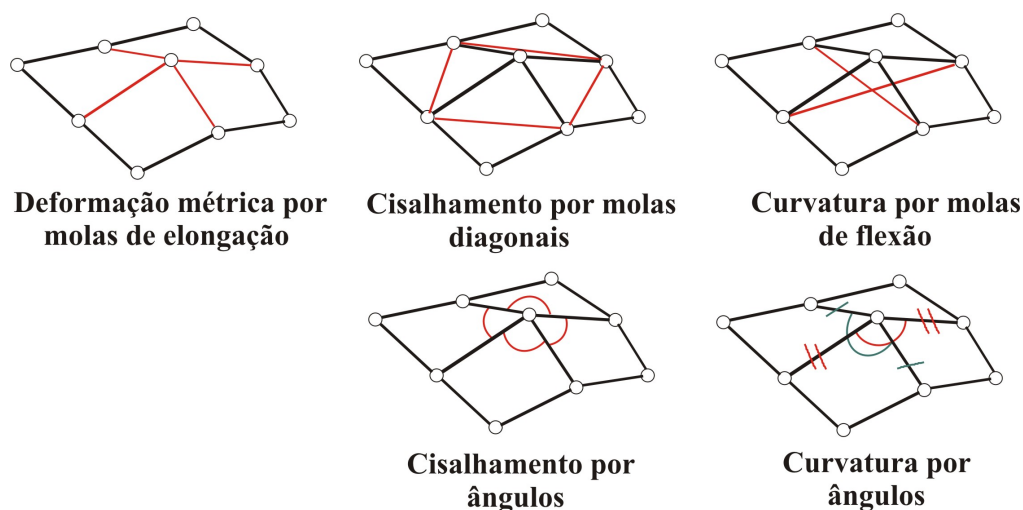


Figura 2.3: Comprimentos de molas ou ângulos versus deformações no sistema de partículas.

2.2 Revisão Bibliográfica

Dentre os primeiros trabalhos com sistemas de partículas na simulação de tecidos temos os de Feynman [20] e de Kunii et al. [29]. O primeiro não contempla o controle de cisalhamento, o que dificulta muito simulações realistas com movimento. Assim ele pode ser visto como um modelo de simulação estática. O segundo possui controles de deformações métricas e de cisalhamento por molas e controle de curvatura por ângulos.

Desenvolvimento substancial na área de modelagem de tecidos com sistemas de partículas encontramos no trabalho de Breen et al. [8]. Esse modelo usa funções energia para o cálculo das deformações baseadas em dados experimentais de tecidos reais obtidos por Kawabata [27]. O processo de simulação ocorre em duas fases: a primeira com as partículas livres sob ação de forças externas como a gravidade e a segunda minimizando as funções energia e corrigindo as velocidades apropriadamente.

Provot [37] introduz, em seu modelo, procedimentos de limitação de deformação melhorando a qualidade dessa classe de modelos. Com um esquema de integração totalmente explícito, esse modelo também tem por característica importante sua eficiência de tempo. Tratamentos de deformação também são abordados por Dochev et al. em [17]. Volino et al. [48] inclui em seu modelo uma resistência à variação de área, característica existente em tecidos reais. Baraff and Witkin usam integração implícita em [2] para uso de malhas com grande número de pontos e passos maiores de tempo. Van Gelder [44] expõe um método para determinação de rigidez em sistemas massa-mola com malha triangular para melhor aproximação das características reais de um tecido.

Dentre os mais recentes trabalhos, Bridson et al. [11] usa malha triangular e desenvolve um sistema para a pré-moldagem de dobras. O trabalho apresenta um esquema de integração misto (parte implícito, parte explícito) eficiente. Choi e Ko [13] trabalham a propriedade de curvatura com atenção especial à formação de dobras e rugas.

Uma visão geral sobre boa parte das primeiras técnicas e métodos de modelagem para tecidos pode ser encontrada no trabalho de Ng e Grimsdale [36]. O livro de House e Breen [26] é rico em informações sobre animação de tecidos além de aspectos como controle e renderização.

Nos modelos de partículas mais antigos observamos nenhum ou um inadequado tratamento para colisões. Dentre eles temos os de Lafleur et al. [30] e de Carignan et al. [12]. O primeiro é mais preciso porém consome mais tempo de processamento. Também não possui otimizações na detecção de colisões. O segundo utiliza um método baseado em dinâmica inversa¹ e leis de fricção macroscópica para a resposta à colisão. Em Volino et al. [47] observamos importantes otimizações para colisões e auto-colisões em modelos de partículas através do uso de hierarquias de volumes envolventes e o uso da curvatura da superfície. Em Provot [38] temos um algoritmo geométrico mais robusto em complemento ao seu modelo de partículas [37]. Encontramos também nesse trabalho otimizações à detecção de colisões e resposta à contato e fricção.

Estruturas hierárquicas de volumes envolventes para detecção de interferência en-

¹Um procedimento de dinâmica inversa é caracterizado e estudado no Capítulo 3 para controle de deformação.

tre objetos são de fundamental importância para a eficiência de algoritmos de colisão [35]. Bons artigos nesses tópicos são os de Gottschalk [23] que estuda OBB trees (caixas envolventes orientadas com o objeto - Oriented Bounding Boxes), o de Van Den Berger [43] usando AABB trees (caixas envolventes orientadas com os eixos coordenados - Axes Aligned Bounding Boxes) e o de Klosowski et al. [28] usando k-DOP's (poliedros convexos envolventes definidos por k semi-espacos - Discrete Orientation Polytope).

Entre trabalhos recentes em detecção e tratamento de colisões podemos citar os de Bridson et al. [10, 11] para malhas retangular e/ou triangular, o artigo de Fuhrmann et al. [21], os três usando detecção geométrica de colisões, e o de Dochev et al. [18], este último usando técnicas de espaço-imagem² para a detecção de colisões.

O livro [46] de Volino e N. Magnenat-Thalmann é uma excelente referência em detecção e tratamento de colisões (e no estudo de modelos deformáveis) e onde são discutidas técnicas e métodos inclusive de renderização. Outro trabalho interessante é o de Gottschalk [24] onde são discutidas diversas técnicas para as mais variadas representações dos objetos.

²Essa técnica gera mapas (ou imagens) de profundidade do objeto (nesse caso um ator virtual) para detectar as colisões.

Modelo Baseado em Malha Retangular

Apresentamos neste capítulo o modelo proposto por Provot [37]. Nele o objeto “tecido” é aproximado por uma superfície deformável composta por uma rede de massas, interligadas por molas, cujo movimento é calculado usando integração numérica da lei fundamental da dinâmica. Este modelo apresenta o problema de super-elasticidade em determinadas circunstâncias. Para corrigir esse “defeito” Provot propõe aplicar um procedimento de correção baseado em dinâmica inversa.

Com o auxílio da nossa implementação, apresentamos diversos testes, submetendo o modelo a vários tipos de forças e situações. Por fim, discutimos os méritos e deméritos dessa proposta.

3.1 O Modelo Massa-Mola

O modelo elástico é composto de uma malha retangular de $m \times n$ massas virtuais interligadas por molas, sem massa, com comprimento natural diferente de zero. Uma *vizinhança* de um ponto $\mathbf{P}_{i,j}$ é o conjunto dos pontos ligados a $\mathbf{P}_{i,j}$ por alguma mola. A ligação entre pontos vizinhos é estabelecida por três tipos de molas (Figura 3.1):

- molas ligando o ponto $[i, j]$ aos pontos $[i + 1, j]$, $[i, j + 1]$, $[i, j - 1]$ e $[i - 1, j]$ são chamadas *estruturais*;
- molas ligando o ponto $[i, j]$ aos pontos $[i + 1, j + 1]$, $[i - 1, j + 1]$, $[i - 1, j - 1]$ e $[i + 1, j - 1]$ são chamadas molas de *cisalhamento*;

- molas ligando o ponto $[i, j]$ aos pontos $[i, j + 2]$, $[i, j - 2]$, $[i - 2, j]$ e $[i + 2, j]$ são chamadas molas de *curvatura*.

Assim, sob deformações puramente estruturais atuarão apenas as molas estruturais, o mesmo acontecendo para as molas de cisalhamento e de curvatura, para deformações puras de suas características.

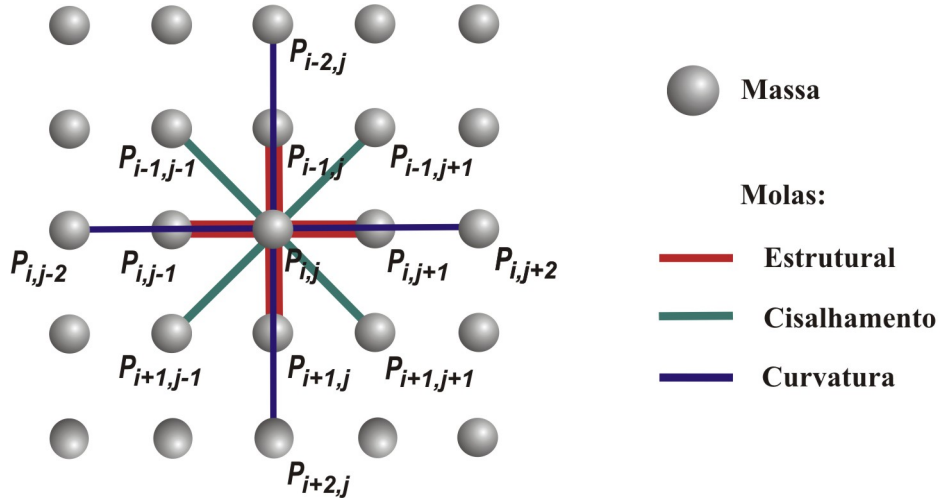


Figura 3.1: Os tipos de molas ligando $\mathbf{P}_{i,j}$ a seus vizinhos.

A distribuição de massa pelos vértices da malha, bem como a distribuição da rigidez das molas, não é diretamente abordada no trabalho original de Provot [37]. Isso nos dá a idéia de que essas distribuições sejam determinadas empiricamente. Em nossa implementação, usamos distribuições de massa e rigidez adotadas por Bridson et al. [11] em seu modelo de malha triangular (descritas no Capítulo 4).

Cada massa está posicionada no tempo t no ponto $\mathbf{P}_{i,j}(t)$, $i = 1, \dots, m$ e $j = 1, \dots, n$, e a evolução do sistema é governada pela 2ª lei de Newton (lei fundamental da dinâmica)

$$\mathbf{F}_{i,j} = \mu_{i,j} \mathbf{a}_{i,j} \quad (3.1)$$

onde $\mu_{i,j}$ é a massa do ponto $\mathbf{P}_{i,j}$ e $\mathbf{a}_{i,j}$ é a aceleração causada pela força $\mathbf{F}_{i,j}$. $\mathbf{F}_{i,j}$ pode ser dividida em força interna e forças externas, isto é,

$$\mathbf{F}_{i,j} = \mathbf{F}_{int} + \mathbf{F}_{ext}. \quad (3.2)$$

A força interna é dada pela resultante das forças exercidas pelas molas que ligam $\mathbf{P}_{i,j}$ a seus vizinhos:

$$\mathbf{F}_{int}(\mathbf{P}_{i,j}) = - \sum_{(k,l)} K_{i,j,k,l} \left[\mathbf{l}_{i,j,k,l} - l_{i,j,k,l}^0 \frac{\mathbf{l}_{i,j,k,l}}{\|\mathbf{l}_{i,j,k,l}\|} \right]$$

onde:

- (k, l) são as coordenadas dos pontos vizinhos a $\mathbf{P}_{i,j}$, ligados por algum tipo de mola (Figura 3.1);
- $\mathbf{l}_{i,j,k,l} = \overrightarrow{\mathbf{P}_{i,j}\mathbf{P}_{k,l}}$;
- $l_{i,j,k,l}^0$ é o comprimento natural da mola entre $\mathbf{P}_{i,j}$ e $\mathbf{P}_{k,l}$;
- $K_{i,j,k,l}$ é a rigidez da mola entre $\mathbf{P}_{i,j}$ e $\mathbf{P}_{k,l}$.

As forças externas são de naturezas variadas de acordo com a situação a qual o modelo é submetido. As mais comuns são a gravidade, amortecimento viscoso (viscosidade do meio em que o corpo está imerso) e a interação viscosa com um fluido (fluido em movimento interagindo com o corpo - comumente o vento). Sendo \mathbf{g} a aceleração da gravidade, o peso de $\mathbf{P}_{i,j}$ é dado por

$$\mathbf{F}_{gr}(\mathbf{P}_{i,j}) = \mu_{i,j}\mathbf{g}.$$

O amortecimento viscoso é dado por

$$\mathbf{F}_{visc}(\mathbf{P}_{i,j}) = -C_{visc}\mathbf{v}_{i,j},$$

onde C_{visc} é o coeficiente de amortecimento e $\mathbf{v}_{i,j}$ a velocidade de $\mathbf{P}_{i,j}$. Esse amortecimento representa uma primeira aproximação de dissipação da energia mecânica do modelo. Por fim, um fluido viscoso movendo-se a uma velocidade uniforme \mathbf{u}_f exerce, em $\mathbf{P}_{i,j}$, a força

$$\mathbf{F}_f(\mathbf{P}_{i,j}) = C_f [\mathbf{n}_{i,j} \cdot (\mathbf{u}_f - \mathbf{v}_{i,j})] \mathbf{n}_{i,j},$$

onde $\mathbf{n}_{i,j}$ é o vetor normal (unitário) à superfície no ponto $\mathbf{P}_{i,j}$ e C_f o coeficiente de viscosidade do fluido. Desse modo, usando as forças anteriormente mencionadas, teremos $\mathbf{F}_{ext} = \mathbf{F}_{gr} + \mathbf{F}_{visc} + \mathbf{F}_f$.

Uma vez computada a força $\mathbf{F}_{i,j}$ (equação (3.2)) em um tempo t a equação (3.1) pode ser explicitamente integrada no tempo pelo método de Euler (Apêndice A)

$$\begin{cases} \mathbf{a}_{i,j}(t + \Delta t) = \frac{1}{\mu_{i,j}} \mathbf{F}_{i,j}(t) \\ \mathbf{v}_{i,j}(t + \Delta t) = \mathbf{v}_{i,j}(t) + \Delta t \mathbf{a}_{i,j}(t + \Delta t) \\ \mathbf{P}_{i,j}(t + \Delta t) = \mathbf{P}_{i,j}(t) + \Delta t \mathbf{v}_{i,j}(t + \Delta t) . \end{cases} \quad (3.3)$$

Este método é conhecido por ser muito rápido e obter bons resultados quando o passo de tempo Δt é menor que o período natural do sistema $T_0 = \pi\sqrt{\frac{\mu}{K}}$. Provot [37], experimentalmente, mostra que a solução numérica do sistema (3.3) é estável para

$$\Delta t \leq 0.4\pi\sqrt{\frac{\mu}{K}}, \quad (3.4)$$

onde μ é a menor massa do sistema e K a maior rigidez. Na Seção 3.4 mostramos a instabilidade do modelo em casos onde a relação (3.4) não é satisfeita.

Em alguns casos o movimento de um tecido não é inteiramente causado por forças analiticamente computáveis como, por exemplo, nos problemas de contato. Uma situação típica desse fato ocorre quando uma cortina encontra-se pendurada por uma vareta. Associemos ao tecido da cortina uma malha de pontos amostrados uniformemente. Cada ponto fixo da cortina (vértice fixo da malha) é puxado para baixo pela gravidade mas a vareta exerce uma força contra-peso que não pode ser calculada como função das posições e velocidades dos vértices da malha. Conhecemos a posição dos pontos fixos (posições constantes ao longo do tempo, no nosso exemplo, posto que a vareta é fixa) e queremos calcular as velocidades e forças resultantes aplicadas aos mesmos (neste caso, ambas são nulas). Tudo ocorre como se não levássemos em conta o resultado da integração do sistema (3.3) para os pontos fixos: quando o resultado for encontrado suas posições são restauradas ao seu valor prévio. Isso caracteriza um *procedimento de dinâmica inversa*. Esses procedimentos também são usados para lidar com auto-colisões do tecido e colisões com outros objetos.

3.2 O Efeito “Super-Elástico”

Uma vez implementado o algoritmo proposto por Provot [37], submetêmo-lo a diversas situações-teste procurando principalmente as ocorrências de falta de realismo e suas causas.

Estudemos aqui o caso (abordado em [37]) de uma bandeira pendurada por dois de seus cantos adjacentes, sujeita a gravidade e sem vento ($\mathbf{u}_f = \mathbf{0}$). Usamos uma malha de 17×17 pontos com os seguintes parâmetros do modelo:

- comprimento natural da mola estrutural de 0.2 cm ;
- rigidez de 0.503 N/m para as molas estruturais, 0.031 N/m para as de curvatura e 0.305 N/m para as de cisalhamento;
- passo de tempo 0.0085 s .

Os dois cantos fixos permanecem imóveis através do procedimento de dinâmica inversa descrito na Seção 3.1. A Figura 3.2(a) mostra a posição das molas estruturais (representadas por linhas) do modelo da bandeira e a Figura 3.2(b) mostra o resultado computado pelo algoritmo após 300 passos.

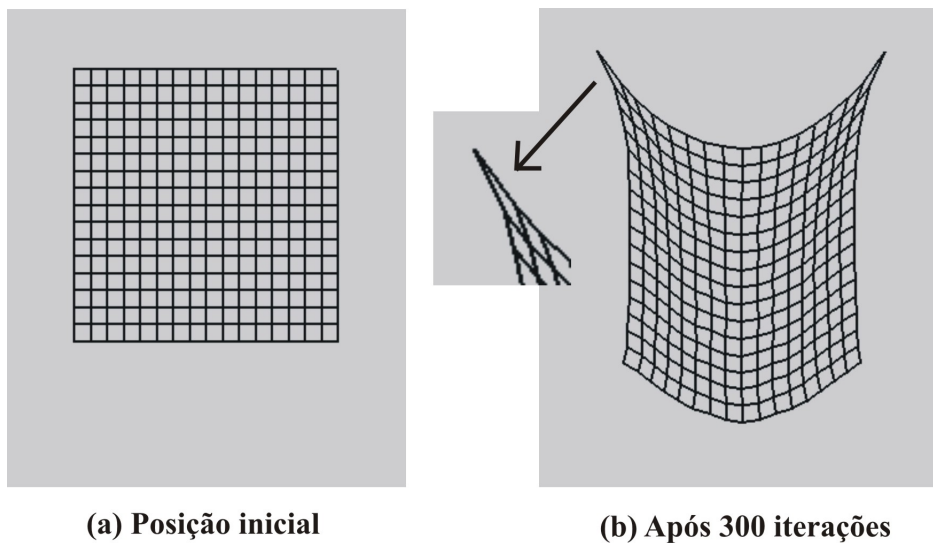


Figura 3.2: Deformação do modelo elástico: bandeira pendurada por dois cantos.

Este exemplo mostra claramente um problema: a elongação das molas diretamente ligadas aos pontos fixos é muito grande comparada às demais. O comprimento dessas molas, nesse exemplo, aumentou para 0.433 cm , ou seja, um crescimento de 116%! Tal deformação local raramente ocorre em tecidos. No campo da física de polímeros ocorre um fenômeno parecido, chamado de “*super-elasticidade*”.

A razão dessa diferença entre o modelo e um tecido real é que o último não é super-elástico em geral e sua elasticidade é não-linear, ocorrendo ruptura antes de grandes

deformações. Não é objetivo do algoritmo modelar rupturas mas limitar deformações indesejáveis. Na maioria dos tecidos a deformação máxima permitida está por volta dos 10% [27], podendo variar um pouco para menos (tecidos como o brim) ou para mais (tecidos como o lycra[®], por exemplo).

A super-elongação não estabiliza facilmente, o que leva a uma grande oscilação em torno do ponto de equilíbrio da bandeira, no exemplo da Figura 3.2. Para resolver esse problema, seria necessário aumentar o coeficiente C_{visc} na equação (3.1) (aumentando o amortecimento da velocidade do objeto ou corpo) mas isso acarreta a aparência de imergir a bandeira em um óleo e o movimento perde realismo.

Aumentar a rigidez das molas poderia ser a solução para o problema da super-elasticidade mas, pela expressão (3.4), teríamos que diminuir o Δt para que o sistema (3.3) não divirja. Isso aumenta consideravelmente o custo do algoritmo. Para um mesmo tempo de animação seriam necessárias um número maior de iterações.

Assim deve-se corrigir o problema sem ter que alterar os parâmetros de viscosidade e passo de tempo. Assunto da seção seguinte.

3.3 Controle da Deformação por Dinâmica Inversa

As deformações podem ser controladas através da aplicação de um procedimento de dinâmica inversa nas molas super-elongadas para reduzir seus comprimentos. A cada passo de tempo, quando as novas posições forem calculadas, calculamos a taxa percentual de deformação em relação ao estado de repouso das molas $\tau = \frac{l - l_0}{l_0}$ e, caso esta seja maior que um valor crítico τ_C (normalmente 0.1, ou seja 10%), o procedimento é aplicado às extremidades da mola de forma que $\tau = \tau_C$.

Neste procedimento, é assumido que a posição da mola, computada pelo sistema (3.3), é correta com respeito à direção mas não com respeito à distância entre suas extremidades. Basta então reduzir essa distância mantendo a mesma direção.

Existem duas situações onde as molas podem estar super-elongadas. Uma quando as extremidades estão soltas e a outra quando uma delas está fixa, por um procedimento de dinâmica inversa. Se ambas estão soltas (Figura 3.3(a)), são “puxadas” em direção ao ponto intermediário da mola de modo que $\tau = \tau_C$. Se uma delas é fixa (Figura 3.3(b)), a extremidade solta é “puxada” na direção da extremidade fixa de modo que $\tau = \tau_C$.

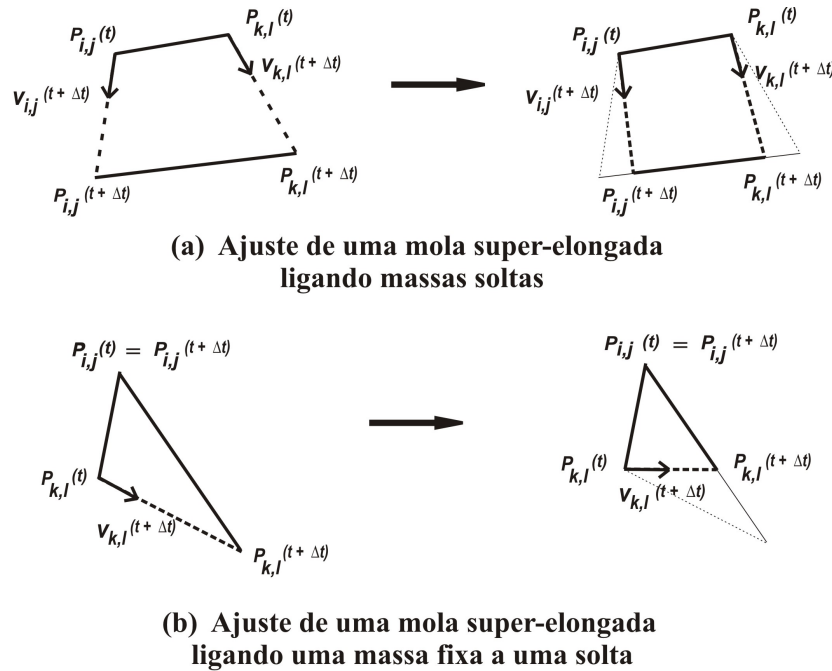
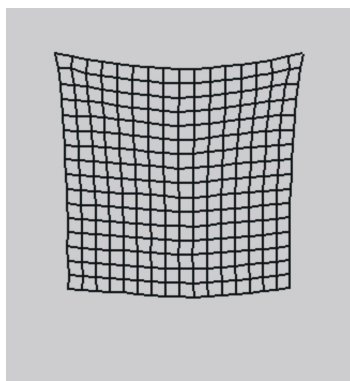


Figura 3.3: Controle da deformação por dinâmica inversa: ajuste de molas super-elongadas.

É claro que essa operação modifica a posição de muitos pontos e pode alongar em excesso outras molas. Mas, se a deformação estiver muito concentrada localmente, as molas afetadas por essa operação serão poucas em relação às detectadas antes dela. Isso faz com que a deformação se propague pela malha em vez de permanecer concentrada em uma pequena área. Uma repetição desse procedimento é então executada até que não existam molas super-elongadas.

O efeito dessa técnica pode ser visto na Figura 3.4, utilizando o exemplo da bandeira descrito anteriormente com os mesmos parâmetros do modelo lá estabelecidos a menos da rigidez de cada tipo de mola, que, devido ao controle da super-elasticidade, pode ser reduzida permitindo o uso de passos maiores de tempo na simulação. Usamos 0.045 N/m para estruturais, 0.015 N/m para as de curvatura e 0.0028 N/m para as molas de cisalhamento e um percentual crítico $\tau_C = 0.1$ para as molas estruturais e de cisalhamento. Não se faz restrições às molas de curvatura já que essa característica não é limitada em tecidos reais - eles dobram facilmente. O coeficiente de amortecimento viscoso pôde ser diminuído evitando o efeito “óleo” no movimento. Também não ocorre grande oscilação em torno do ponto de equilíbrio.

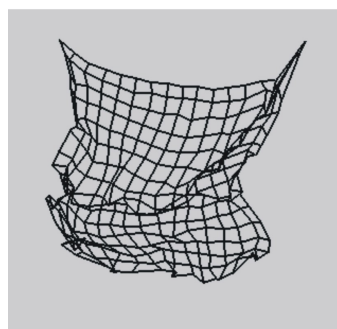


Após 300 iterações

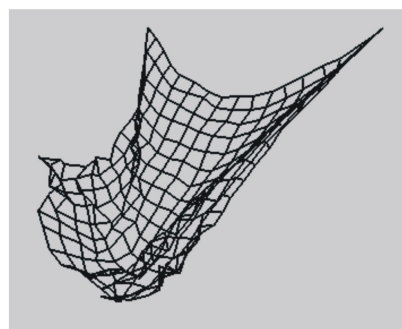
Figura 3.4: Correção das molas super-elongadas.

3.4 Resultados e Discussões

Uma das mais importantes características desse algoritmo é a sua velocidade. O esquema de integração e os procedimentos de dinâmica inversa são simples e de custo computacional muito baixo se comparados aos procedimentos e esquemas usados em modelos elásticos clássicos. Entretanto, não se pode usar grandes passos de tempo com integração explícita pois isso gera instabilidade (Figura 3.5(a)). Essa mesma instabilidade ocorre aumentando a rigidez das molas (Figura 3.5(b)).



(a) $\Delta t = 0.02$



**(b) Rigidez estrutural
K = 2.502 N/m**

Figura 3.5: Instabilidade no exemplo da Seção 3.2.

A correção do efeito super-elástico não leva em conta a ordem com que as molas super-elongadas são ajustadas em cada passo de tempo. Ela depende inteiramente da estrutura de dados usada na implementação. Em nosso trabalho, a correção é feita por varredura das linhas da malha $m \times n$. Provot atenta para o fato de que essa ordem

pode fazer diferença caso as super-elongações sejam muitas e distribuídas pelo objeto. A abordagem desse problema foge ao escopo do nosso trabalho e pode ser encontrada em [17]. Em nossa implementação, por simplicidade, usamos no procedimento iterativo da correção da super-elasticidade um número fixo de iterações. Quatro iterações foram suficientes para bons resultados visuais.

O procedimento de dinâmica inversa, por atuar na posição dos pontos, acrescenta energia ao sistema e, sob determinadas circunstâncias (configurações dos parâmetros do modelo), afeta a sua dinâmica. Na Figura 3.6, vemos a bandeira com um deslocamento irreal para um dos lados já que está apenas sob o efeito da gravidade. Percebe-se também que a bandeira não entra em estado de equilíbrio, pois, há uma oscilação constante na posição dos pontos o que aparenta um “tremor” durante a animação.

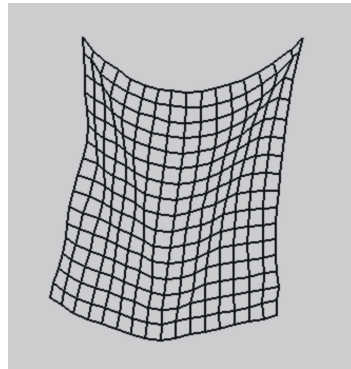


Figura 3.6: Energia indevida produzida pela correção da posição.

Este modelo também não gera dobras e rugas! Considere o caso de uma bandeira pendurada, como no exemplo da Seção 3.3, com todos os pontos dispostos num plano vertical de mesma direção da gravidade (sem vento). A partir de um instante qualquer “soltamos” um dos pontos fixos. A bandeira cai, ficando presa por um único ponto fixo. Na Figura 3.7(a), mostramos o que acontece nessa situação.

Na Figura 3.7(b), usamos o artifício de distribuir os pontos de maneira não planar e, dessa forma, já que existem componentes de forças internas na direção normal, são geradas as rugas e dobras. No Capítulo 7 apresentamos uma proposta para resolver tal problema sem o artifício apresentado aqui.

No algoritmo (Provot [37]), originalmente, não é feito o tratamento para as intercolisões ou colisões, como atualmente se costuma fazer. Por isso, a estabilidade e preservação das dobras não são mencionadas.

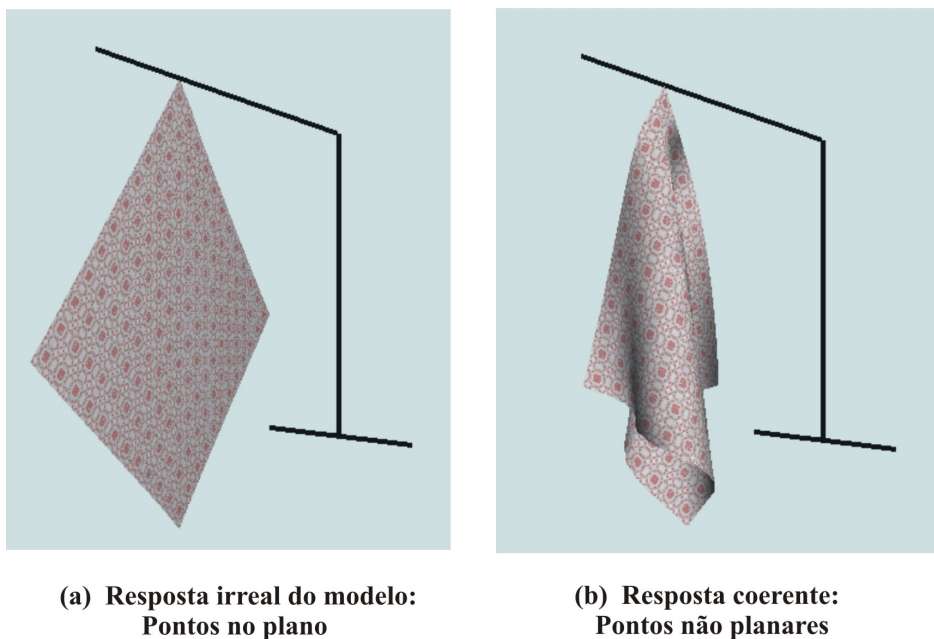


Figura 3.7: Modelo versus dobras espontâneas.

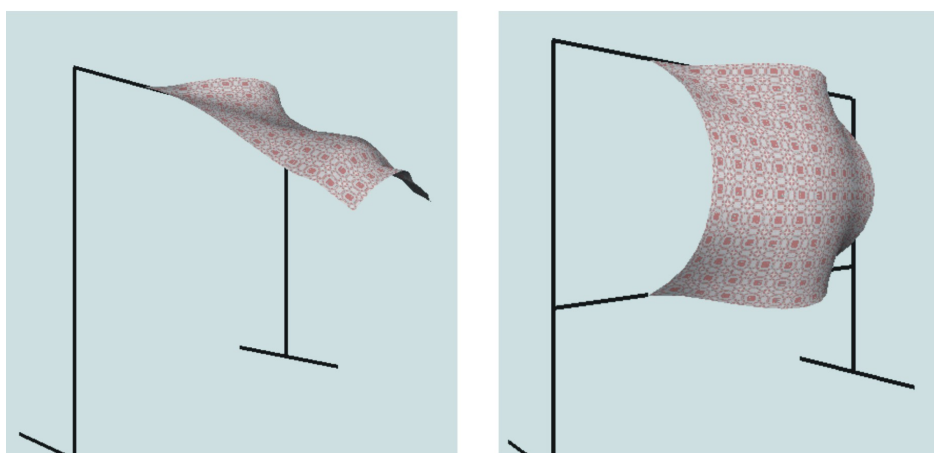


Figura 3.8: Mais exemplos: bandeira e vela ao vento.

A Figura 3.8 ilustra o funcionamento do modelo em duas situações semelhantes às encontradas no trabalho original de Provot. Tanto nesse exemplo, Figura 3.8, quanto no anterior, Figura 3.7, usamos uma textura aplicada sobre o tecido. O uso de texturas aproxima o visual do tecido modelado com o tecido real.

Capítulo 4

Modelo Baseado em Malha Triangular

Apresentamos neste capítulo o modelo proposto por Bridson et al. [11]. Diferentemente do modelo estudado no capítulo anterior, este não possui molas de curvatura. Em vez disso, usa para modelar a curvatura o ângulo diedral dos pares de triângulos que têm uma aresta comum.

Uma particularidade desse modelo é o uso de um esquema misto de integração no tempo, ou seja, o sistema que governa o seu movimento é integrado parte explicitamente e parte implicitamente. Outra particularidade é o uso de um procedimento de correção da super-elasticidade mais coerente e apropriado para o (futuro) tratamento de colisões.

Complementamos esse estudo com a implementação do algoritmo, testes, análise e discussões sobre o modelo.

4.1 O Modelo Massa-Mola

Este modelo usa uma malha de massas ligadas por molas estruturais, como na Seção 3.1, onde cada quadrado possui apenas uma mola de cisalhamento em uma de suas diagonais. Além disso as molas diagonais são alternadas de quadrado para quadrado, o que nos leva a um padrão “espinha de peixe” (Figura 4.1). Isso tem por objetivo não introduzir tendências para as deformações na malha, como certamente ocorreria no caso de usarmos a mesma diagonal em todos os quadrados. Note que cada vértice interior da malha possui seis graus de liberdade para deslocamentos.

A distribuição de massa nos vértices da malha pode ser feita¹ calculando-se a

¹Existem maneiras mais complicadas, como o uso de células de Voronoi. Esse estudo não faz parte do escopo do nosso trabalho. As distribuições que usamos já bastam para nossos objetivos.

massa de cada triângulo, através da densidade de massa por área do tecido, e atribuindo um terço dessa massa a cada vértice do triângulo. Se um vértice pertence a mais de um triângulo receberá a massa correspondente de cada um deles. Os autores utilizam rigidez $K = m/l_0$ para as molas, onde m é a massa do vértice e l_0 o comprimento de repouso da mola, com finalidade de tornar a velocidade de onda $l_0\sqrt{K/ml_0}$ independente dos parâmetros da malha. Em nosso trabalho usamos a maior rigidez estrutural, assim calculada, para todas as molas estruturais e, analogamente, a maior rigidez de cisalhamento para todas as molas de cisalhamento.

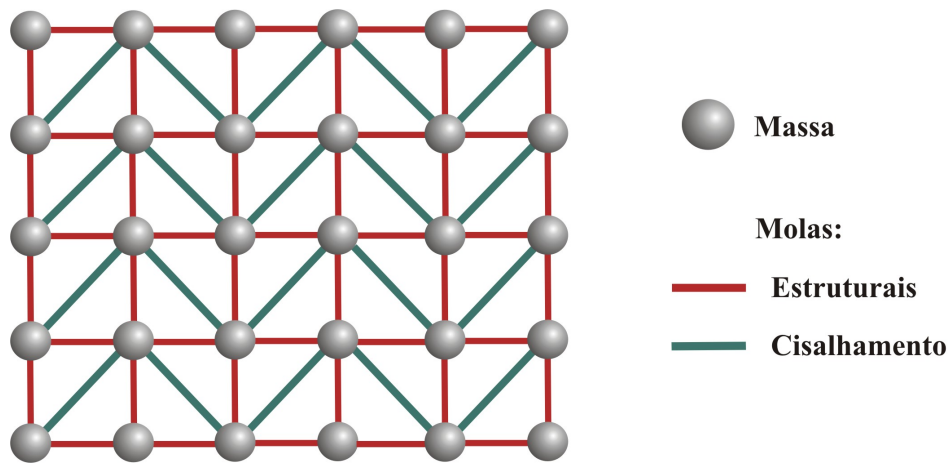


Figura 4.1: Malha triangular.

As forças internas e externas atuantes no sistema são semelhantes às descritas no capítulo anterior, Seção 3.1. A diferença reside na ausência de molas de curvatura. As forças de curvatura do modelo serão obtidas através de elementos de curvatura.

4.1.1 Elementos de Curvatura

Chamamos *elemento de curvatura* a um par de triângulos da malha que têm uma aresta comum. O elemento opera baseado no ângulo diedral, donde obtemos a componente *elástica* da força de curvatura, e em sua taxa de variação, donde obtemos a componente *amortecimento* para a força de curvatura². A Figura 4.2 ilustra um elemento de curvatura com vértices de posições \mathbf{x}_i , velocidades \mathbf{v}_i , $i = 1, \dots, 4$, e ângulo θ entre as normais \mathbf{n}_1 e \mathbf{n}_2 .

O vetor $\mathbf{F} = (\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_4)$ das quatro forças de curvatura (elásticas ou de amor-

²técnica usada por Baraff e Witkin [2]

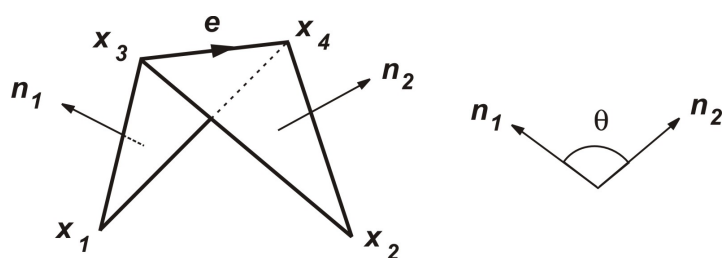


Figura 4.2: Elemento de curvatura com ângulo diedral $\pi - \theta$.

tecimento) pertence a um espaço linear 12-dimensional. Podemos selecionar uma base para esse espaço identificando doze “modos” de movimento. Temos três translações, três rotações (instantâneas), dois movimentos no plano (do triângulo $\mathbf{x}_1\mathbf{x}_3\mathbf{x}_4$) do vértice \mathbf{x}_1 , dois movimentos no plano (do triângulo $\mathbf{x}_2\mathbf{x}_3\mathbf{x}_4$) do vértice \mathbf{x}_2 e uma deformação da aresta $\mathbf{x}_3 - \mathbf{x}_4$ em sua reta suporte (Figura 4.3).

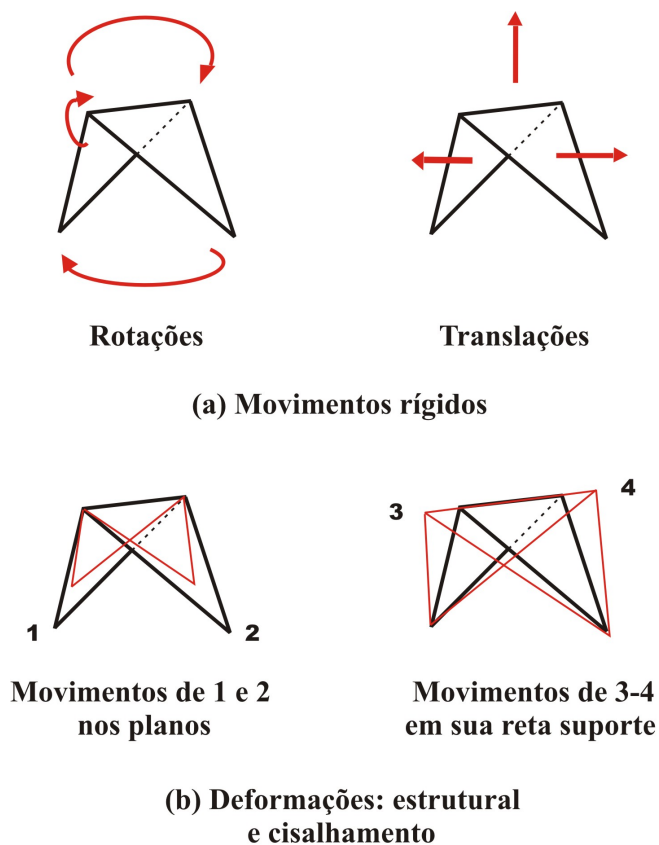


Figura 4.3: Modos de movimento para o elemento de curvatura.

Nenhum desses onze modos alteram o ângulo diedral e portanto não devem participar dos cálculos das forças de curvatura. Nos restou o décimo segundo modo, o *modo de curvatura*, que é ortogonal a todos os outros. Este modo altera o ângulo diedral mas

não causa nenhuma deformação ou movimento rígido ao elemento. Vamos chamá-lo de $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4)$. Pela condição de ortogonalidade aos movimentos no plano dos vértices \mathbf{x}_1 e \mathbf{x}_2 (não gerar movimento dos vértices nos respectivos planos) temos que \mathbf{u}_1 é paralelo a \mathbf{n}_1 e \mathbf{u}_2 é paralelo a \mathbf{n}_2 . Pela condição de ortogonalidade à deformação da aresta $\mathbf{x}_3 - \mathbf{x}_4$ sobre sua reta suporte (não gerar deformações nesta aresta ao longo de sua reta suporte), obtemos que $\mathbf{u}_4 - \mathbf{u}_3$ está no espaço gerado por \mathbf{u}_1 e \mathbf{u}_2 (plano ortogonal à aresta $\mathbf{x}_3 - \mathbf{x}_4$). A ortogonalidade com as translações (não gerar translações) nos diz que $\mathbf{u}_1 + \mathbf{u}_2 + \mathbf{u}_3 + \mathbf{u}_4 = \mathbf{0}$. Assim, $\mathbf{u}_3 + \mathbf{u}_4$ também pertencem ao espaço gerado por \mathbf{u}_1 e \mathbf{u}_2 , implicando que \mathbf{u}_3 e \mathbf{u}_4 pertencem a esse espaço. Por fim, para fazer \mathbf{u} ortogonal às rotações (convenientemente escolhidas como sendo em torno dos eixos \mathbf{n}_1 , \mathbf{n}_2 e \mathbf{e}), Bridson et al. [11] sugerem

$$\begin{aligned} \mathbf{u}_1 &= |\mathbf{E}| \frac{\mathbf{N}_1}{|\mathbf{N}_1|^2} & \mathbf{u}_2 &= |\mathbf{E}| \frac{\mathbf{N}_2}{|\mathbf{N}_2|^2} \\ \mathbf{u}_3 &= \frac{[(x_1 - x_4) \cdot \mathbf{E}]}{|\mathbf{E}|} \frac{\mathbf{N}_1}{|\mathbf{N}_1|^2} + \frac{[(x_2 - x_4) \cdot \mathbf{E}]}{|\mathbf{E}|} \frac{\mathbf{N}_2}{|\mathbf{N}_2|^2} \\ \mathbf{u}_4 &= -\frac{[(x_1 - x_3) \cdot \mathbf{E}]}{|\mathbf{E}|} \frac{\mathbf{N}_1}{|\mathbf{N}_1|^2} - \frac{[(x_2 - x_3) \cdot \mathbf{E}]}{|\mathbf{E}|} \frac{\mathbf{N}_2}{|\mathbf{N}_2|^2}, \end{aligned} \quad (4.1)$$

onde $\mathbf{N}_1 = (\mathbf{x}_1 - \mathbf{x}_3) \times (\mathbf{x}_1 - \mathbf{x}_4)$ e $\mathbf{N}_2 = (\mathbf{x}_2 - \mathbf{x}_4) \times (\mathbf{x}_2 - \mathbf{x}_3)$ são normais (carregando informações das áreas dos triângulos) e $\mathbf{E} = \mathbf{x}_4 - \mathbf{x}_3$ a aresta comum. Portanto, \mathbf{u}_1 e \mathbf{u}_2 são inversamente proporcionais as distâncias dos vértices \mathbf{x}_1 e \mathbf{x}_2 à aresta comum e \mathbf{u}_3 e \mathbf{u}_4 são combinações lineares de \mathbf{u}_1 e \mathbf{u}_2 . As forças de curvatura devem ser proporcionais a esse modo.

É interessante observar que, pela ortogonalidade aos modos de movimentos rígidos, o modo de curvatura conserva o momento linear e o momento angular do elemento. De fato, qualquer modelo de curvatura baseado em dois triângulos que não use essas direções para suas forças vai violar um e/ou outro momento ou causará deformações indesejáveis.

A magnitude da componente elástica foi escolhida como sendo

$$\mathbf{F}_i^e = K^e \frac{|\mathbf{E}|^2}{|\mathbf{N}_1| + |\mathbf{N}_2|} \text{sen}(\theta/2) \mathbf{u}_i, \quad i = 1, \dots, 4. \quad (4.2)$$

A rigidez de curvatura elástica K^e é uma propriedade do material do tecido e independente da malha e o fator $\text{sen}(\theta/2)$ mede o quão “distante” de um plano o elemento de curvatura está. Usa-se a fórmula $\text{sen}(\theta/2) = \pm \sqrt{(1 - \mathbf{n}_1 \cdot \mathbf{n}_2)/2}$ (visto que

$\mathbf{n}_1 \cdot \mathbf{n}_2 = \cos\theta$) onde o sinal escolhido deve ser o mesmo que o do $\sin\theta$ que, por sua vez, tem o mesmo sinal que $\mathbf{n}_1 \times \mathbf{n}_2 \cdot \mathbf{e}$, com $\mathbf{e} = \frac{\mathbf{E}}{|\mathbf{E}|}$. O fator central³ $\frac{|\mathbf{E}|^2}{|\mathbf{N}_1|+|\mathbf{N}_2|}$ ajusta-se conforme a propriedade anisotrópica da malha de modo que alterações em sua estrutura (refinamento, por exemplo) não afetem significativamente o funcionamento do modelo. Uma interpretação geométrica desse fator pode ser obtida com o auxílio da Figura 4.4.

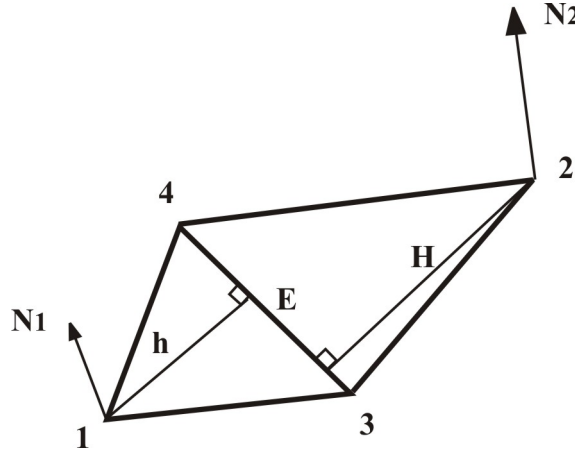


Figura 4.4: Interpretação geométrica do fator $\frac{|\mathbf{E}|^2}{|\mathbf{N}_1|+|\mathbf{N}_2|}$.

Como a área do paralelogramo formado pelos vetores $\vec{14}$ e $\vec{13}$ (esse paralelogramo tem por diagonais a aresta **34** e o vetor $\vec{14} + \vec{13}$) é o dobro da área do triângulo **134** temos que

$$\frac{1}{2} |\mathbf{N}_1| = \frac{|\mathbf{E}| h}{2}$$

e portanto $h = |\mathbf{N}_1| / |\mathbf{E}|$, sendo $\mathbf{N}_1 = \vec{13} \times \vec{14}$. Com o mesmo raciocínio sobre o triângulo **234** chegamos a $H = |\mathbf{N}_2| / |\mathbf{E}|$, com $\mathbf{N}_2 = \vec{24} \times \vec{23}$, e dessas duas últimas expressões obtemos

$$\frac{|\mathbf{E}|}{H + h} = \frac{|\mathbf{E}|^2}{|\mathbf{N}_1| + |\mathbf{N}_2|},$$

ou seja, o fator central é diretamente proporcional ao comprimento da aresta **34** e inversamente proporcional à soma das distâncias dos vértices **1** e **2** a esta mesma aresta.

O modelo pode ser incrementado com a funcionalidade de simular pré-dobras no tecido, como vincos em uma calça, através de ângulos de repouso diferentes de zero. Isso pode ser feito pela força elástica

$$\mathbf{F}_i^e = K^e \frac{|\mathbf{E}|^2}{|\mathbf{N}_1| + |\mathbf{N}_2|} (\sin(\theta/2) - \sin(\theta_0/2)) \mathbf{u}_i, \quad i = 1, \dots, 4,$$

³Este fator é obtido empiricamente.

onde θ_0 é o ângulo de repouso. Isso introduz uma força interna inicial para curvar nas regiões onde se deseja dobras ou rugas. A Figura 4.5(a), retirada de [11], mostra essa interessante capacidade. Na parte (b), da mesma figura, vemos a deformação do modelo submetido ao próprio peso. Não utilizaremos essa funcionalidade em nosso trabalho.

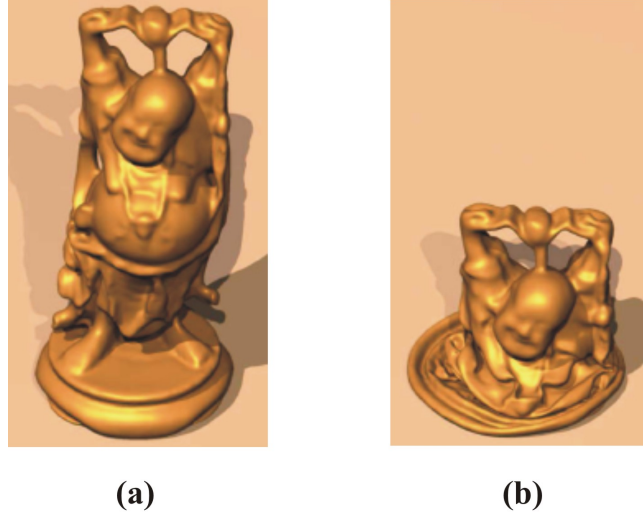


Figura 4.5: Ângulos de repouso permitem pré-esculpir dobras nos tecidos.

A componente de amortecimento da força de curvatura depende da taxa de variação do ângulo diedral, atuando de modo a diminuí-la. Bridson et al. [11] sugerem a aproximação

$$d\theta/dt = \mathbf{u}_1 \cdot \mathbf{v}_1 + \mathbf{u}_2 \cdot \mathbf{v}_2 + \mathbf{u}_3 \cdot \mathbf{v}_3 + \mathbf{u}_4 \cdot \mathbf{v}_4 \quad (4.3)$$

para $d\theta/dt$, evidenciando a estreita ligação entre as velocidades dos vértices e as direções do modo de curvatura na variação do ângulo. Uma interpretação geométrica dessa expressão pode ser vista na Figura 4.6 (visão planar do elemento de curvatura) tomando os vértices 1 e 2 como exemplo. Na parte (a) não há variação do ângulo diedral ao passo que em (b) o ângulo varia “proporcionalmente” aos produtos internos $\mathbf{u}_1 \cdot \mathbf{v}_1$ e $\mathbf{u}_2 \cdot \mathbf{v}_2$.

Assim, a componente de amortecimento será dada por

$$\mathbf{F}_i^d = -K^d |\mathbf{E}| (d\theta/dt) \mathbf{u}_i, \quad i = 1, \dots, 4,$$

onde novamente foi incluído um fator, $|\mathbf{E}|$, para que a propriedade do material K^d possa permanecer constante mediante alterações da malha.

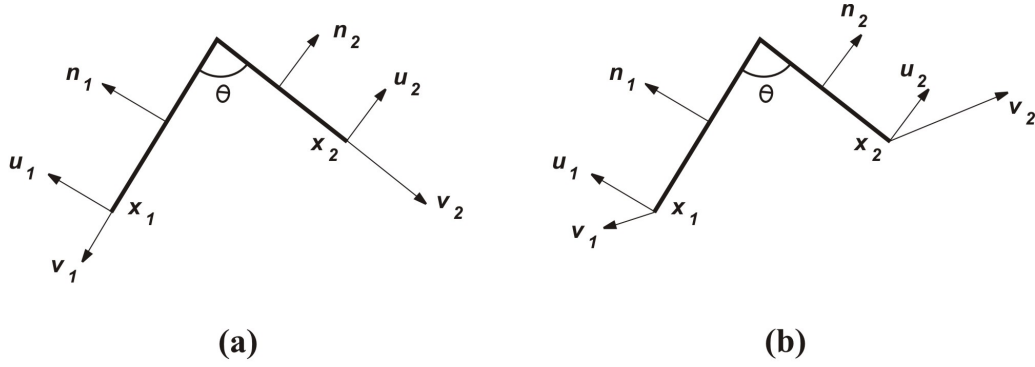


Figura 4.6: Interpretação geométrica para a aproximação (4.3) de $d\theta/dt$.

Finalmente temos, para cada vértice i do elemento, $i = 1 \dots 4$, a força de curvatura

$$\mathbf{F}_i^{cur} = \mathbf{F}_i^e + \mathbf{F}_i^d .$$

4.2 Integração Mista

O modelo usa um esquema misto de integração no tempo (da 2ª lei de Newton, $\mathbf{F} = m\mathbf{a}$) combinando a flexibilidade e simplicidade dos métodos explícitos com a eficiência dos métodos implícitos. Em particular, é usada a integração explícita nas forças elásticas (aquelas que não dependem da velocidade) e integração implícita sobre as forças de amortecimento⁴ (aquelas que dependem da velocidade). O uso da atualização explícita da velocidade permite, alterando velocidades de maneira simples, impor restrições à dinâmica do modelo como, por exemplo, o controle da deformação para a super-elasticidade, descrito mais adiante.

O algoritmo é apresentado a seguir, onde \mathbf{x} denota posição, \mathbf{v} velocidade e \mathbf{a} aceleração:

- $\mathbf{v}^{n+1/2} = \mathbf{v}^n + \frac{\Delta t}{2} \mathbf{a}(t^n, \mathbf{x}^n, \mathbf{v}^n);$ (explícito)
- Modifica-se $\mathbf{v}^{n+1/2}$, para controle da deformação, obtendo $\tilde{\mathbf{v}}^{n+1/2};$
- $\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t \tilde{\mathbf{v}}^{n+1/2};$ (explícito)

⁴Bridson et al. [11] consideram como força de amortecimento qualquer uma que dependa da velocidade. As forças fluido \mathbf{F}_f , a viscosidade \mathbf{F}_{visc} e o amortecimento para a curvatura \mathbf{F}_i^d fazem parte dessa definição.

- $\mathbf{v}^{n+1} = \mathbf{v}^{n+1/2} + \frac{\Delta t}{2} \mathbf{a}(t^{n+1}, \mathbf{x}^{n+1}, \mathbf{v}^{n+1});$ (implícito)
- Modifica-se \mathbf{v}^{n+1} (nela mesma) para controle da deformação.

Essencialmente, o algoritmo combina um esquema explícito de segunda ordem, para atualizar posições, com um esquema implícito de segunda ordem, o método trapezoidal⁵, para atualizar velocidades. Dessa forma, este é um método de segunda ordem de precisão e mantém-se estável para

$$\Delta t < \mathcal{O}(\Delta x \sqrt{\rho/K^e}), \quad (4.4)$$

onde ρ é a densidade de massa, K^e a rigidez elástica do material e Δx o menor comprimento de aresta da malha.

É sugerido um esquema de integração alternativo para materiais muito “amortecidos”. Nesses casos o primeiro algoritmo sugerido, embora estável, não é monótono. Isto é, oscilações espúrias podem ser introduzidas pelo meio-passo explícito. A abordagem utilizada para essa situação é atualizar as posições com um meio-passo implícito e, em seguida, aplicar o método trapezoidal:

- $\tilde{\mathbf{v}}^{n+1/2} = \mathbf{v}^n + \frac{\Delta t}{2} \mathbf{a}(t^n, \mathbf{x}^n, \mathbf{v}^{n+1/2});$ (implícito)
- Modifica-se $\tilde{\mathbf{v}}^{n+1/2}$, (nela mesma) para controle da deformação;
- $\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t \tilde{\mathbf{v}}^{n+1/2};$ (explícito)
- $\mathbf{v}^{n+1/2} = \mathbf{v}^n + \frac{\Delta t}{2} \mathbf{a}(t^n, \mathbf{x}^n, \mathbf{v}^n);$ (explícito)
- $\mathbf{v}^{n+1} = \mathbf{v}^{n+1/2} + \frac{\Delta t}{2} \mathbf{a}(t^{n+1}, \mathbf{x}^{n+1}, \mathbf{v}^{n+1});$ (implícito)
- Modifica-se \mathbf{v}^{n+1} (nela mesma) para controle da deformação.

Isso permite o uso de passos de tempo variáveis, precisão de segunda ordem e comportamento monótono, às custas de mais uma solução implícita.

4.3 Controle da Deformação

Embora o procedimento de controle da deformação proposto por Provot [37], descrito no capítulo anterior, pareça funcionar bem, envolve movimento de vértices e

⁵Isso pode ser visto substituindo $\mathbf{v}^{n+1/2}$ da primeira linha do algoritmo na última.

pode causar auto-interseções na malha. Daí sua inadequação para o uso com algoritmos de colisões. O procedimento aqui proposto ajusta velocidades em vez de posições, evitando esse inconveniente.

A cada passo de tempo, calculamos o candidato a novo comprimento da mola entre duas partículas usando as velocidades correntes. Caso a deformação seja excessiva, ajustamos as velocidades, conservando o momento linear do conjunto, de modo que a deformação permitida seja de, no máximo, 10% do seu comprimento de repouso, ao final do passo de tempo. A Figura 4.7 ilustra esse método. Na parte (a) o (futuro) comprimento da mola extrapolou seu limite, na parte (b) os impulsos foram aplicados e em (c) são mostrados os momentos inicial e final (os mesmos).

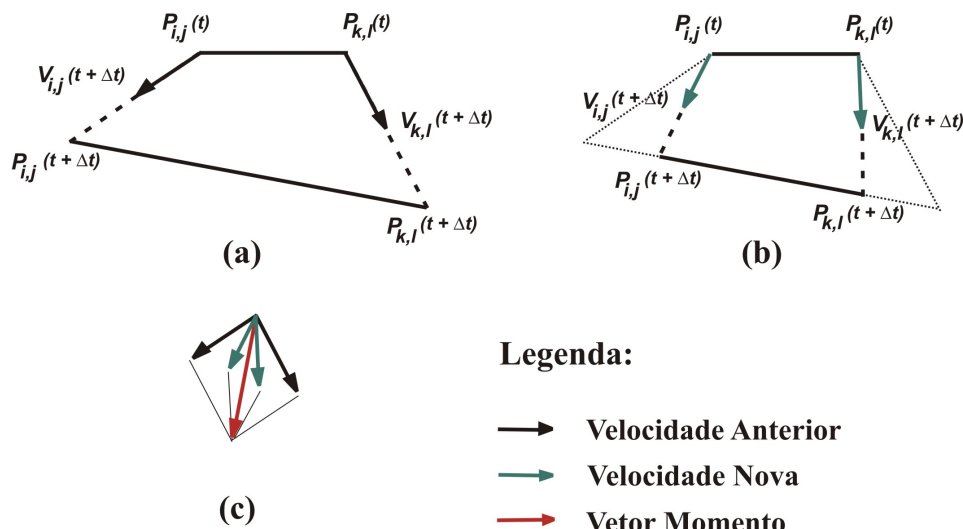


Figura 4.7: Controle da deformação: ajuste das velocidades com conservação do momento linear.

O ajuste é realizado sequencialmente, isto é, as velocidades são atualizadas imediatamente e podem afetar os impulsos aplicados às velocidades de vértices vizinhos. Torna-se necessário, então, um procedimento iterativo para assegurar que nenhuma mola deforme mais que 10% de seu comprimento natural dentro de um mesmo passo de tempo. Embora essas iterações não possuam convergência garantida, apenas uma ou duas são suficientes para resultados visualmente agradáveis.

Outra propriedade usada por esse modelo é a restrição da compressão a 0%, ou seja, desabilitando-a completamente. Isto é feito para aumentar o realismo no tocante à formação de dobras e rugas. Aplicando impulsos que bloqueiam a compressão planar

do tecido permite-se uma maior curvatura para fora do plano⁶, facilitando a ocorrência de dobras ou rugas. De fato, na tentativa de comprimir um tecido (real) obtemos, invariavelmente, seu enrugamento em vez de sua compressão.

4.4 Controle da Taxa de Deformação

Em computação gráfica, podem ocorrer grandes velocidades em modelos fisicamente baseados. Um exemplo disso ocorre em animação quando um ator, desenhado à mão, se move com grande (e fisicamente incorreta) aceleração, levando consigo o tecido. Isso geralmente acarreta uma taxa de deformação muito alta em um passo de tempo. Na física computacional costuma-se limitar essa taxa em 10%.

A maneira natural de controlar a taxa de deformação é reduzir adaptativamente o passo de tempo, garantindo um limite para a taxa. Entretanto, isso tornaria a simulação lenta e um número maior de passos de tempo seriam necessários para um mesmo tempo de animação. Uma alternativa atraente, proposta por Bridson et al. [10], consiste em, iterativamente, aplicar impulsos que reduzam a taxa de deformação (por passo de tempo) a um limite aceitável de maneira análoga à usada para controle da deformação. Dessa forma, podemos continuar usando, continuamente, passos de tempo largos sem prejuízos ao tempo de execução.

4.5 Resultados e Discussões

O primeiro ponto positivo desse modelo está na própria malha. Enquanto que o uso de uma malha retangular é simples para a simulação de um pedaço de tecido (como bandeiras, cortinas, roupas simples, etc), não o é para objetos de geometria mais complicada. A malha triangular se adapta melhor à descrição de objetos mais complexos em sua geometria. Além disso, malhas triangulares representam de forma mais adequada superfícies curvas (como as dobras e rugas) o que torna esse modelo muito mais realístico, quando comparado a um de malha retangular.

Os controles de deformação e taxa de deformação por impulsos aplicados às velocidades (momento-conservativos) são coerentes, robustos e eficientes. Acrescentam pouca energia ao sistema e, por isso, não geram oscilações indesejáveis.

⁶A Seção 4.5 mostra que isso não ocorre caso os vértices estejam todos em um mesmo plano.

O custo computacional deste procedimento excede o do usado por Provot [37], visto que aqui há duas fases de controle (da deformação e da taxa de deformação).

O esquema misto de integração é mais preciso que um puramente explícito e mais simples (computacionalmente) que um puramente implícito. Em nossa implementação, por simplicidade, usamos no cômputo da etapa implícita da integração apenas as forças de amortecimento viscoso do ar e do fluido, excluindo as forças de curvatura. Esse argumento nos permite calcular facilmente essa etapa sem o uso de um sistema linear. De fato, observe que a velocidade nessa etapa será dada por (veja a Seção 3.1)

$$\mathbf{v}_{i,j}^{n+1} = \mathbf{v}_{i,j}^n + dt(1/m_{i,j})(-C_{visc}\mathbf{v}_{i,j}^{n+1} + C_f [\mathbf{n}_{i,j} \cdot (\mathbf{u}_f - \mathbf{v}_{i,j}^{n+1})] \mathbf{n}_{i,j})$$

ou ainda, melhor escrevendo

$$\mathbf{v}_{i,j}^{n+1} = \mathbf{v}_{i,j}^n + dt(1/m_{i,j})(-C_{visc}\mathbf{v}_{i,j}^{n+1} + C_f(\mathbf{n}_{i,j} \cdot \mathbf{u}_f)\mathbf{n}_{i,j} - C_f(\mathbf{n}_{i,j} \cdot \mathbf{v}_{i,j}^{n+1})\mathbf{n}_{i,j}).$$

Observe ainda que conhecendo $(\mathbf{n}_{i,j} \cdot \mathbf{v}_{i,j}^{n+1})$ teremos uma equação de solução fácil para $\mathbf{v}_{i,j}^{n+1}$. Isso pode ser feito calculando o produto interno por $\mathbf{n}_{i,j}$, obtendo

$$(\mathbf{v}_{i,j}^{n+1} \cdot \mathbf{n}_{i,j}) = (\mathbf{v}_{i,j}^n \cdot \mathbf{n}_{i,j}) + dt(1/m_{i,j})(-C_{visc}(\mathbf{v}_{i,j}^{n+1} \cdot \mathbf{n}_{i,j}) + C_f(\mathbf{n}_{i,j} \cdot \mathbf{u}_f) - C_f(\mathbf{n}_{i,j} \cdot \mathbf{v}_{i,j}^{n+1}))$$

de onde isolamos $(\mathbf{v}_{i,j}^{n+1} \cdot \mathbf{n}_{i,j})$. Note que, se usássemos as forças de curvatura apareceriam nesses cálculos as velocidades de diversos pontos e os mesmos não seriam assim tão simples. Deixamos então ao cargo da etapa explícita o cômputo do efeito do amortecimento das forças de curvatura.

A Figura 4.8 mostra um exemplo de uma bandeira pendurada por dois cantos consecutivos, exatamente como na Seção 3.2 do capítulo anterior. Neste exemplo, utilizamos os mesmos parâmetros (ao menos os comuns a ambos os modelos): comprimento natural da mola estrutural de 0.2 cm, velocidade do vento $\mathbf{u}_f = \mathbf{0}$, malha de 17×17 vértices, 220 passos de tempo de 0.02 s, rigidez das molas estruturais e de cisalhamento de 0.01 N/m e 0.0071 N/m, respectivamente, rigidez elástica e de amortecimento de 0.0077 N/m e 0.00035 N/m, respectivamente, deformação e taxa de deformação máximas permitidas de 10% e 5%, respectivamente.

Observamos, nesse simples exemplo, muitas das características do modelo já citadas nessa seção. A dobra surge facilmente e aparenta mais realismo em sua forma. A animação apresenta-se “suave”, atingindo um estado de equilíbrio perfeito. Uma

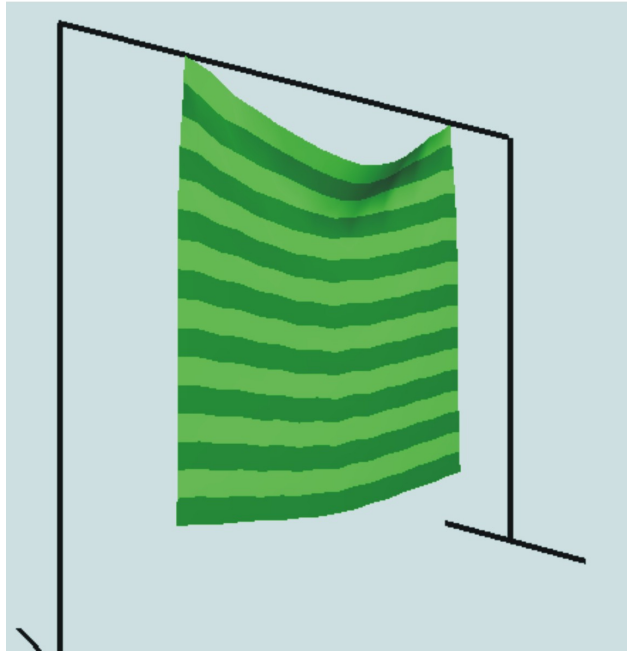


Figura 4.8: Deformação do modelo elástico: bandeira pendurada por dois cantos.

característica não evidente antes de nossa implementação aparece na baixa rigidez necessária para as molas estruturais e de cisalhamento. O realismo não é prejudicado por esses baixos valores como ocorre no modelo proposto por Provot [37]. Isso nos permitiu o uso do passo de tempo maior e, portanto, menos iterações foram necessárias para animação (no exemplo do capítulo anterior usamos 300 iterações e, mesmo assim, não atingimos o equilíbrio).

A Figura 4.9 mostra o exemplo da bandeira anterior submetida ao vento. Na parte (a) com o modelo baseado em malha retangular do Capítulo 3 e na parte (b) com o modelo de malha triangular. Nota-se a diferença de realismo nos resultados. A aparência e suavidade de forma e, principalmente, de movimentos do tecido é superior no modelo de malha triangular. O Apêndice C exhibe alguns quadros da animação de cada uma dessas bandeiras para melhor observação da dinâmica de ambos os modelos.

Este modelo, como o descrito no capítulo anterior, também não gera dobras e rugas espontaneamente! Considere a mesma situação exposta na Seção 3.4. Uma bandeira pendurada com todos os pontos dispostos num plano vertical de mesma direção da gravidade (sem vento). A partir de um instante qualquer “soltamos” um dos pontos fixos. A bandeira cai, ficando presa por um único ponto fixo.

A Figura 4.10(a), ilustra o que acontece nessa situação. Nas Figuras 4.8 e 4.10(b),

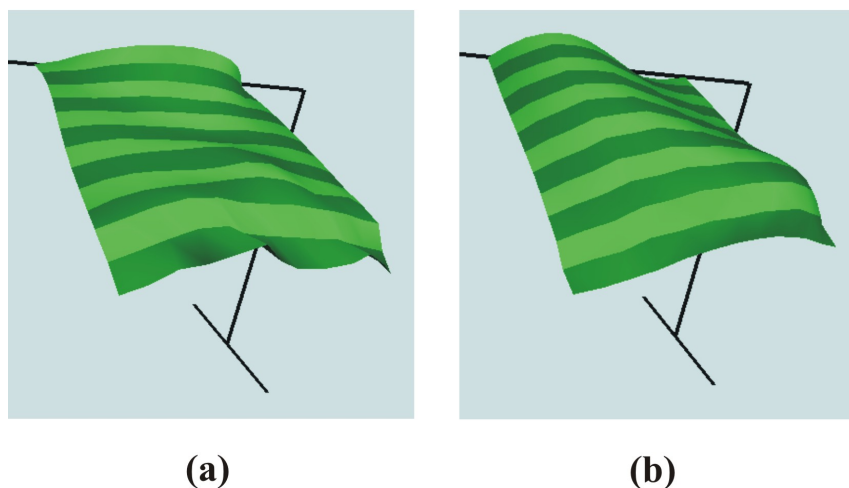


Figura 4.9: Bandeiras ao vento em modelos distintos: (a) malha retangular; (b) malha triangular.

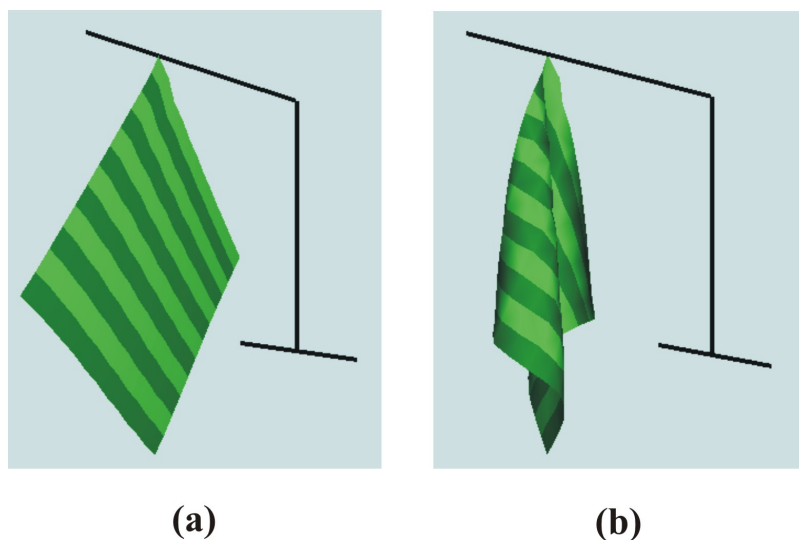


Figura 4.10: O modelo não gera dobras espontaneamente: (a) pontos distribuídos no plano; (b) pontos com distribuição não planar.

usamos o artifício de distribuir os pontos de maneira não planar e, dessa forma, já que existem componentes de forças internas na direção normal, são geradas as rugas e dobras. Como já mencionamos, no Capítulo 7 apresentamos uma proposta para resolver tal problema sem o uso desse artifício.

A faculdade de pré-moldagem de dobras e/ou rugas (mencionada na Seção 4.1.1) desse modelo não é possível de ser implementada no modelo de malha retangular. Se uma posição de repouso curvada (posição de equilíbrio) for usada para determinar os comprimentos de repouso das molas, a posição curva do tecido na direção oposta (como

sua imagem no espelho) também é uma posição de equilíbrio, pois determina os mesmos comprimentos. Dessa forma, os vértices podem oscilar entre essas duas posições durante a animação “saltando” entre elas. Isso descaracteriza a pré-moldagem.

O trabalho original de Bridson et al. [11], como é do feitio dos trabalhos mais recentes da área, trata da estabilidade e preservação das dobras e rugas através de um algoritmo para colisões com objetos. Em nosso trabalho, preferimos deixar tratamentos de colisões (sejam inter-colisões ou com outros objetos) em capítulos separados.

A Figura 4.11 traz mais um exemplo de situação para o modelo da bandeira, idêntica a apresentada ao final do Capítulo 3 para o modelo de malha retangular.

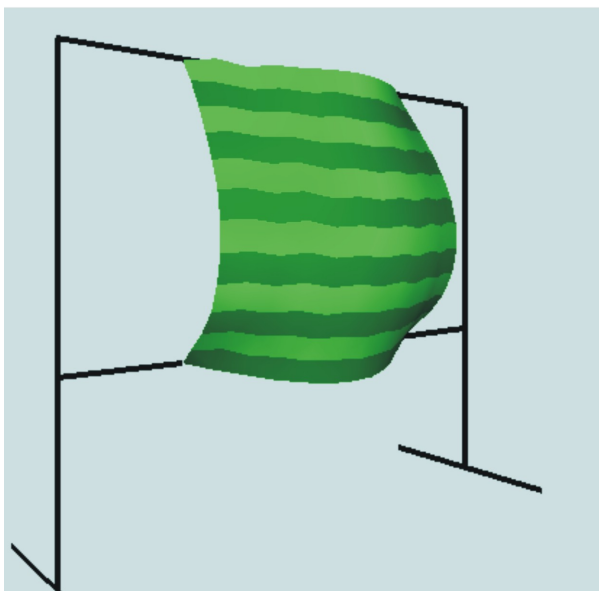


Figura 4.11: Mais um exemplo para o modelo de malha triangular: vela ao vento.

Algoritmo de Colisão Para o Modelo de Malha Retangular

Este capítulo apresenta o algoritmo para o tratamento de colisões, do tecido com objetos e consigo próprio, proposto por Provot [38] para o seu modelo de malha retangular [37], descrito no Capítulo 3.

Dentre as principais dificuldades no tratamento de colisões podemos citar a detecção, a otimização da detecção, a resposta à colisão e a consistência, ou seja, a sua estabilidade ao longo da dinâmica dos objetos envolvidos. O algoritmo em questão lida com cada uma dessas dificuldades. É apresentado então um método robusto para o cômputo da resposta à múltiplas colisões.

Por fim, discutimos detalhes da implementação, as características do algoritmo e as impressões obtidas através de nossas simulações.

5.1 Detecção de Colisões

A detecção de colisões se faz necessária entre o objeto deformável (tecido) e objetos móveis de uma cena e, em particular, consigo mesmo. Levando-se em consideração a detecção e a resposta à colisão, ambos os casos são tratados da mesma forma. A diferença existe na otimização para as auto-colisões, como oportunamente será discutido. Assim, não faremos distinção entre auto-colisões e colisões entre objetos diferentes.

O elemento de colisão básico nesse algoritmo é o triângulo, portanto todos os objetos usados numa cena devem ser aqui representados por eles, inclusive o nosso modelo deformável do tecido. Para o modelo de malha retangular [37] pode-se, por exemplo,

tomar uma diagonal em cada quadrado, obtendo uma representação triangulada do tecido.

Seja t_0 um instante em que não hajam penetrações entre o tecido e o objeto. Conhecendo as posições e velocidades de cada vértice do modelo em t_0 é possível computar suas posições em $t_0 + \Delta t$. A detecção de colisões consiste em verificar, no intervalo $[t_0, t_0 + \Delta t]$, se uma (ou várias) das seguintes situações ocorreu (ou ocorreram):

- um vértice de uma malha toca o triângulo da outra (colisão ponto-triângulo);
- uma aresta do triângulo de uma malha toca uma aresta de um triângulo da outra (colisão aresta-aresta).

É importante atentar para o fato de que, no método de Euler usado pelo modelo, durante o intervalo $[t_0, t_0 + \Delta t]$, é considerado que cada vértice se move a uma velocidade constante. Essa propriedade é fundamental para o algoritmo, como veremos nas seções seguintes.

5.1.1 A Colisão Ponto-Triângulo

Seja $\mathbf{P}(t)$ um ponto em movimento e $\mathbf{A}(t)$, $\mathbf{B}(t)$ e $\mathbf{C}(t)$ os vértices de um triângulo, também em movimento. Sejam \mathbf{V} , \mathbf{V}_A , \mathbf{V}_B e \mathbf{V}_C suas respectivas velocidades constantes durante o intervalo $[t_0, t_0 + \Delta t]$. Assim, temos $\mathbf{P}(t) = \mathbf{P}(t_0) + t\mathbf{V}$, $\mathbf{A}(t) = \mathbf{A}(t_0) + t\mathbf{V}_A$, $\mathbf{B}(t) = \mathbf{B}(t_0) + t\mathbf{V}_B$ e $\mathbf{C}(t) = \mathbf{C}(t_0) + t\mathbf{V}_C$.

Na ocorrência de uma colisão, deverá existir t no intervalo $[t_0, t_0 + \Delta t]$ tal que $\mathbf{P}(t)$ pertença ao triângulo $\mathbf{ABC}(t)$ (Figura 5.1), ou seja

$$\exists t \in [t_0, t_0 + \Delta t] \text{ tal que}$$

$$\exists u, v \in [0, 1], u + v \leq 1, \mathbf{AP}(t) = u\mathbf{AB}(t) + v\mathbf{AC}(t) \quad (5.1)$$

Como esta equação vetorial nos leva a um sistema não linear, precisamos de outra condição que expresse o fato de que o ponto pertença ao triângulo. Ocorre que, o produto vetorial $\mathbf{N}(t) = \mathbf{AB}(t) \times \mathbf{AC}(t)$ é perpendicular ao plano do triângulo \mathbf{ABC} e, portanto, a relação

$$\mathbf{AP}(t) \cdot \mathbf{N}(t) = 0 \quad (5.2)$$

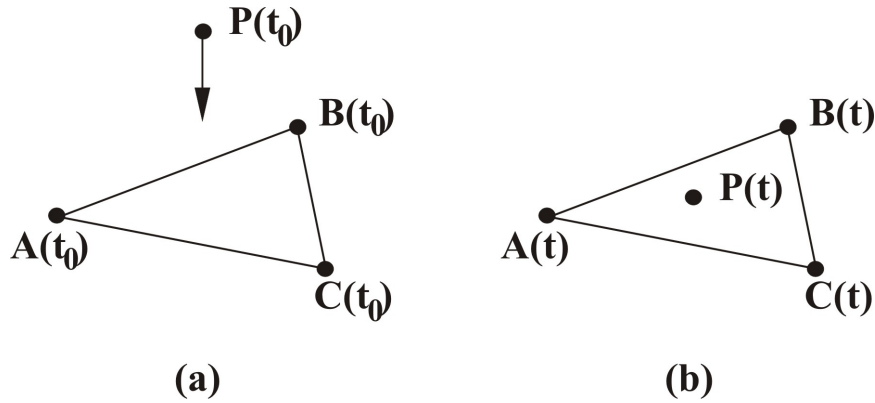


Figura 5.1: Colisão ponto-triângulo: (a) antes da colisão; (b) durante a colisão.

deve ser satisfeita no tempo de colisão.

Esta última relação é necessária mas não suficiente. O ponto $\mathbf{P}(t)$ pode pertencer ao plano que contém o triângulo mas não estar contido no triângulo. Não obstante, ela nos permitirá determinar o tempo de colisão de maneira direta. De fato, $\mathbf{N}(t)$ é uma expressão em t^2 e $\mathbf{AP}(t)$ uma expressão em t , portanto a relação (5.2) é uma equação de terceiro grau de fácil solução. Dos três valores obtidos, apenas os que estão no intervalo $[t_0, t_0 + \Delta t]$ podem corresponder a uma colisão. Para identificar quais desses valores identificam uma colisão, e não coplanaridades, os injetamos de volta na equação (5.1) tornando-a um sistema linear.

Se vários valores (t, u, v) forem soluções do sistema, considera-se como colisão aquele que ocorreu primeiro, ou seja, o que possui o menor valor de t .

5.1.2 A Colisão Aresta-Aresta

Sejam $\mathbf{AB}(t)$ uma aresta do tecido e $\mathbf{CD}(t)$ uma aresta de um objeto em movimento (como havíamos dito, o objeto pode ser o próprio tecido). Na ocorrência de uma colisão deverá existir $t \in [t_0, t_0 + \Delta t]$ tal que (Figura 5.2)

$$\exists u, v \in [0, 1], \mathbf{A}(t) + u(\mathbf{B} - \mathbf{A})(t) = \mathbf{C}(t) + v(\mathbf{D} - \mathbf{C})(t) \quad (5.3)$$

Novamente somos levados a um sistema não linear e necessitamos de mais uma relação para obtermos possíveis valores de t . Durante a colisão, os quatro pontos devem estar em um mesmo plano, donde a relação

$$(\mathbf{AB}(t) \times \mathbf{CD}(t)) \cdot \mathbf{AC}(t) = 0$$

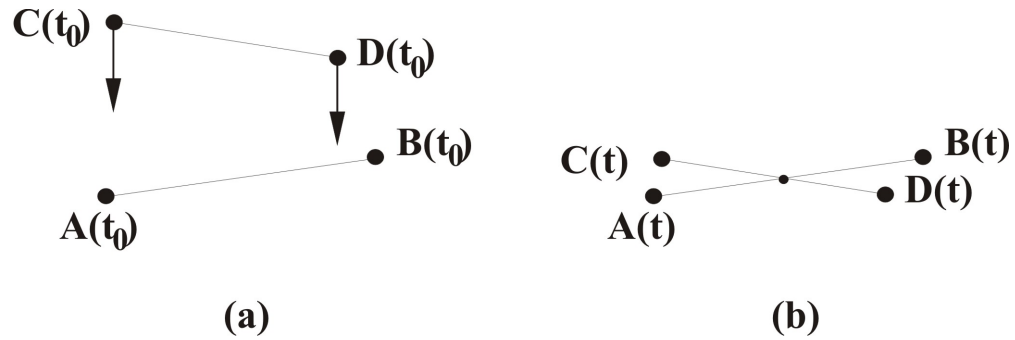


Figura 5.2: Colisão aresta-aresta: (a) antes da colisão; (b) durante a colisão.

deve ser satisfeita nesse instante.

Essa relação se traduz em uma equação de terceiro grau que, da mesma forma que na colisão ponto-triângulo, nos permite calcular u e v após injetar os valores de t na equação (5.3). A solução (t, u, v) também será a que ocorreu primeiro.

5.2 Otimizações da Detecção de Colisões

No tratamento de colisões, a detecção de colisão é a que mais consome tempo de processamento. Por esse motivo, um algoritmo força bruta¹ sobre os procedimentos de detecção, estudados na seção anterior, torna-se inviável para malhas de tecido e de objetos com grande número de pontos. Assim, otimizações são necessárias. As hierarquias de volumes envolventes e a otimização pela curvatura da superfície são as mais comuns.

5.2.1 Hierarquia de Caixas Envolventes

Este tipo de otimização, já bem conhecida, consiste na divisão recursiva do tecido (e também do objeto) em zonas sobrepostas e no armazenamento dessas zonas numa estrutura de árvore (Figura 5.3). Em nossa implementação, usamos árvores binárias².

A cada iteração, uma caixa envolvente em cada zona é calculada e armazenada junto a ela nos nodos da árvore. O algoritmo de colisão analisa as árvores do tecido e do objeto eliminando testes de colisões desnecessários entre primitivas que estão em caixas com interseção vazia. Para maior precisão do método, as caixas devem envolver

¹Aquele que testa colisões entre todas as primitivas possíveis sem nenhum tipo de otimização.

²Explicação detalhada sobre árvores pode ser encontrado no livro de Loudon [32], com implementações inclusive.

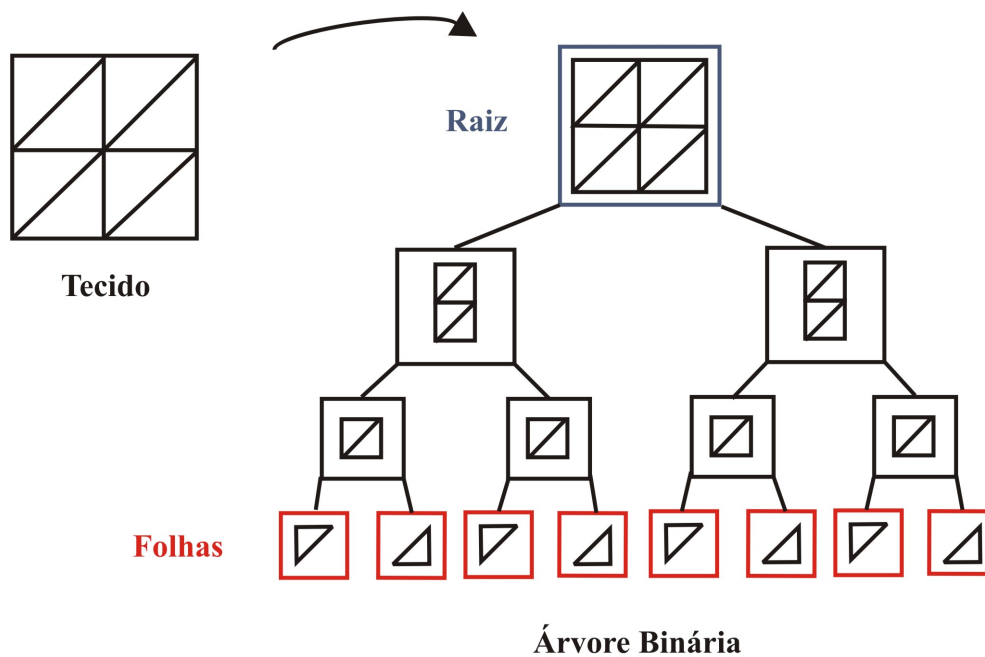


Figura 5.3: Divisão recursiva do tecido em zonas e armazenamento numa árvore binária.

as posições da zona em t_0 e $t_0 + \Delta t$. Em nossa implementação, escolhemos as caixas alinhadas com os eixos (Figura 5.4) ou, usando a terminologia mais conhecida, AABB³ (Axes Aligned Bounding Boxes).

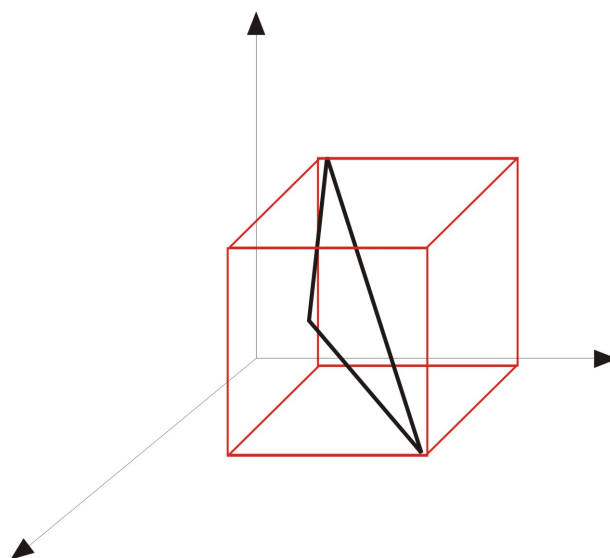


Figura 5.4: Caixa alinhada com os eixos envolvendo uma primitiva.

³Um bom trabalho sobre AABB trees pode ser visto no trabalho de Van Den Bergen [43].

5.2.2 A Curvatura da Superfície e a Detecção de Auto-colisões

A técnica aqui descrita é uma adaptação da proposta por Volino e N. Magnenat-Thalmann [47]. A otimização baseia-se na propriedade de que uma zona (conexa⁴) do tecido com “baixa curvatura” não se auto-intersecta e, é claro, zonas menores dentro dela não se intersectam umas com as outras.

A curvatura de uma zona pode ser avaliada pelo conjunto de normais de cada primitiva que compõe a mesma. Calcula-se um cone com todos esses vetores normais e o ângulo α de seu vértice determinará quando uma zona se auto-intersecta ou não (Figura 5.5). Caso $\alpha < \pi$ a zona não possui auto-colisões.

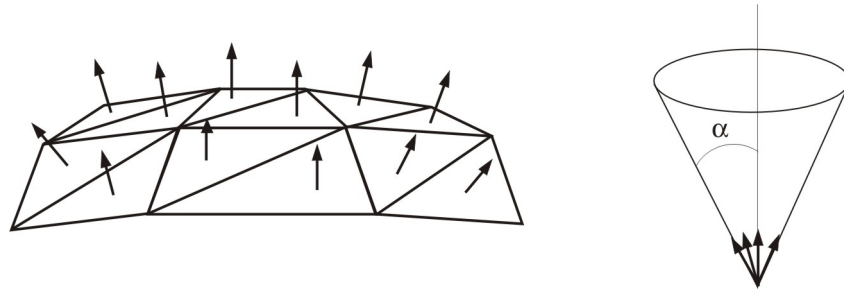


Figura 5.5: Cone de normais para uma zona.

Os cones são calculados usando a estrutura hierárquica da seção anterior. Na base da árvore, cada folha possui um vetor normal e então, nesse tipo de zona, temos $\alpha = 0$ e o eixo desse cone é o próprio vetor normal. Assim, para cada nodo da árvore, do qual conhecemos os cones de seus nodos descendentes, calculamos o cone usando os ângulos α_1 e α_2 de seus descendentes e o ângulo β entre os eixos desses cones. O vetor eixo do novo cone é calculado como a média dos eixos dos cones descendentes e o novo ângulo α (Figura 5.6) é calculado pela expressão

$$\alpha = \beta/2 + \max(\alpha_1, \alpha_2).$$

Provot [38] atenta para o fato de que esse procedimento é válido apenas para cones descendentes cujas zonas sejam adjacentes, o que pode não ocorrer caso o contorno do tecido não for convexo. A estrutura hierárquica que usamos satisfaz esse critério para todos os casos abordados nesse trabalho.

⁴Essa condição, segundo Volino, não é suficiente. Provot ressalta que ela o é, na prática, para os casos necessários ao modelo.

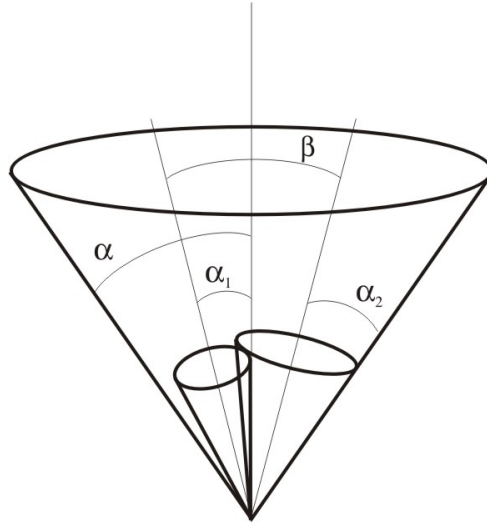


Figura 5.6: Cone de ângulo α calculado a partir dos cones descendentes de ângulos α_1 e α_2 .

5.3 Resposta à Colisão

A resposta à colisão entre dois objetos tem por objetivo primordial evitar interpenetrações entre ambos, mas também simula os efeitos de fricção e de repique (como ocorre quando uma bola cai sobre uma mesa e fica “pulando” sobre ela).

5.3.1 Contato e Fricção

Quando dois objetos colidem, existe um tempo em que eles estão em contato. As leis de fricção macroscópicas de Coulomb [5] podem ser aplicadas nessa situação. Provot [38], em seu trabalho, considera nessa seção colisões perfeitamente inelásticas, ou seja, toda a energia é dissipada e não há o efeito repique.

Seja \mathbf{P} um ponto de massa em contato com uma superfície rígida, sem movimento, no ponto \mathbf{H} e \mathbf{N} o vetor unitário normal à superfície nesse ponto. Seja \mathbf{F} a força aplicada a \mathbf{P} para manter o contato. Assim, $\mathbf{F}_N = (\mathbf{F} \cdot \mathbf{N})\mathbf{N}$ e $\mathbf{F}_T = \mathbf{F} - \mathbf{F}_N$ são as componentes perpendicular e tangencial, respectivamente, da força \mathbf{F} . Pelas leis de fricção, obtemos

- se $\|\mathbf{F}_T\| > k_f \|\mathbf{F}_N\|$, haverá contato deslizante e com fricção, ou seja, o ponto se moverá paralelamente à superfície sob ação da força $\mathbf{F}_S = \mathbf{F}_T - k_f \|\mathbf{F}_N\| \frac{\mathbf{F}_T}{\|\mathbf{F}_T\|}$;
- se $\|\mathbf{F}_T\| \leq k_f \|\mathbf{F}_N\|$, haverá um contato não deslizante e o ponto permanecerá parado, isto é, $\mathbf{F}_S = 0$;

onde $k_f \in \mathbb{R}^+$ é chamado de coeficiente de fricção. Se $k_f = 0$, haverá deslizamento sem fricção e se $k_f = \infty$ não haverá deslizamento algum. O coeficiente de fricção é um parâmetro do tecido e, portanto, deve ser especificado como tal.

Numa colisão ponto-triângulo onde o triângulo esteja parado, a força de contato gerada pelo ponto no triângulo (e vice-versa) é desconhecida. Conhecendo apenas a velocidade \mathbf{V} do ponto, antes da colisão, a sua aceleração pode ser aproximada por $(\mathbf{V}' - \mathbf{V})/\Delta t$, onde \mathbf{V}' é a velocidade do ponto após a colisão. Assim, a força de contato pode ser calculada por $\mathbf{F}_C = m(\mathbf{V}' - \mathbf{V})/\Delta t$. Porém, \mathbf{V}' é justamente o que o algoritmo procura determinar.

Para resolver esse problema é tomada uma aproximação no cálculo da força gerada pelo impacto. Essa aproximação consiste em considerar que as forças envolvidas numa colisão são proporcionais às velocidades, o que, intuitivamente, faz sentido: com uma grande velocidade de impacto, uma grande força é gerada. Assim, qualquer que seja esse coeficiente de proporcionalidade, as leis de fricção anteriormente descritas podem ser reescritas substituindo-se \mathbf{F} por \mathbf{V} e \mathbf{F}_S por \mathbf{V}' .

Essas relações nos darão a velocidade do ponto imediatamente após a colisão. Como a velocidade do ponto é considerada constante durante o intervalo $[t_0, t_0 + \Delta t]$, o algoritmo substitui a velocidade \mathbf{V} do ponto no intervalo por \mathbf{V}' e calcula a sua trajetória. Isso equivale a considerar que a colisão ocorreu exatamente em t_0 , embora o tempo de colisão $t \in [t_0, t_0 + \Delta t]$ tenha sido calculado no processo de detecção.

5.3.2 Impacto e Dissipação

Em contraste com o contato, outro fenômeno durante a colisão é o impacto e seu efeito colateral de repique. Em colisões elásticas não há dissipação de energia, em colisões inelásticas existe dissipação e em colisões perfeitamente inelásticas a energia é inteiramente dissipada.

Esses comportamentos podem ser expressos pela seguinte relação empírica. Com a mesma notação anterior, a velocidade de um ponto \mathbf{P} colidindo com um triângulo parado será, após a colisão, $\mathbf{V}' = \mathbf{V}_T - k_d \mathbf{V}_N$, onde $k_d \in [0, 1]$ é o coeficiente de dissipação (se $k_d = 0$ temos uma colisão perfeitamente inelástica e se $k_d = 1$ temos uma colisão elástica). Esse coeficiente também faz parte dos parâmetros mecânicos do tecido.

5.3.3 Resposta Total

Determinaremos agora a resposta total aplicada aos elementos envolvidos numa colisão. No caso geral, a velocidade $\mathbf{V} = \mathbf{V}_T + \mathbf{V}_N$ de um ponto antes da colisão com um triângulo parado será, após a colisão, substituída por \mathbf{V}' , sendo que

- se $\|\mathbf{V}_T\| > k_f \|\mathbf{V}_N\|$, então $\mathbf{V}' = \mathbf{V}_T - k_f \|\mathbf{V}_N\| \frac{\mathbf{V}_T}{\|\mathbf{V}_T\|} - k_d \mathbf{V}_N$;
- se $\|\mathbf{V}_T\| \leq k_f \|\mathbf{V}_N\|$, então $\mathbf{V}' = -k_d \mathbf{V}_N$.

No caso de objetos móveis, essas relações serão aplicadas à velocidades calculadas no referencial móvel de velocidade igual ao centro de massa do objeto, ou seja, a velocidade \mathbf{V} de cada ponto, usada no cálculo da resposta, será dada pela diferença entre a velocidade real do ponto e a velocidade do centro de massa do objeto. No caso de auto-colisões, as velocidades serão calculadas no referencial de velocidade do centro de massa de todos os elementos envolvidos na colisão; o ponto e o triângulo ou as duas arestas. Isso significa dizer que a resposta atua como se a colisão ocorresse no centro de massa dos elementos envolvidos.

Para a decomposição da velocidade inicial de cada ponto em suas componentes normal e tangencial, a normal usada será a normal do triângulo, no caso ponto-triângulo, ou o produto vetorial (normalizado) das duas arestas, no caso aresta-aresta.

5.4 Consistência para Múltiplas Colisões

Este algoritmo trata o problema de colisões entre pares de elementos, ponto e triângulo e entre duas arestas, de forma independente. Cada cômputo de colisão modifica as posições dos pontos envolvidos e, portanto, dos triângulos e arestas ligados a estes. Nada garante, então, que essas modificações de posições não acarretem novas colisões. Provot [38] denomina estes casos por *múltiplas colisões*.

O método para manter a consistência das colisões usado pelo algoritmo baseia-se na determinação de zonas onde as colisões múltiplas ocorrem e no tratamento específico das mesmas sob uma nova hipótese de colisão.

5.4.1 Zonas de Impacto

Para identificar se, após a detecção e resposta de colisões, novas situações de colisão foram criadas, faz-se uma nova detecção. Caso não sejam detectadas novas colisões o algoritmo avança para o próximo passo de tempo. Caso contrário, seria possível computar uma nova resposta às colisões e repetir o processo até que novas colisões não ocorressem. Entretanto esse procedimento iterativo não possui garantias de convergência.

A primeira fase do método proposto consiste em circunscrever todos os pontos envolvidos em múltiplas colisões. Usando o procedimento iterativo descrito acima, memoriza-se, em cada iteração, os pontos que estejam “unidos” por pertencerem a uma mesma colisão, ponto-triângulo ou aresta-aresta, ou por tomarem parte em colisões diferentes mas com pontos comuns. Esses conjuntos de pontos são denominados *zonas de impacto*.

Em cada iteração, estas zonas podem crescer quando novas colisões surgirem (Figura 5.7). Durante esse processo de crescimento, se duas zonas possuírem pontos comuns serão unidas numa maior. O processo termina quando todas as zonas pararem de crescer, ficando estáveis. Diremos que as zonas estão *circunscritas*.

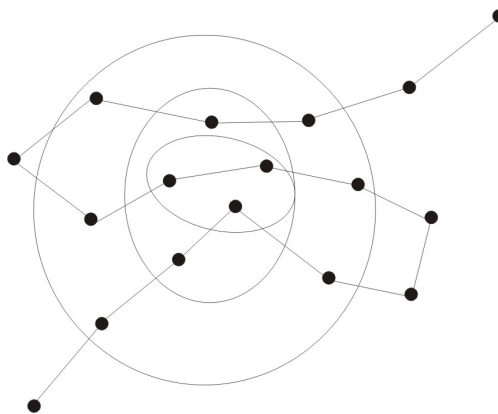


Figura 5.7: Circunscrição iterativa de zonas de impacto.

A convergência desse método é garantida pois, no pior caso, as zonas crescerão e serão unidas até envolverem todo o tecido.

5.4.2 Hipótese de Colisão para Zonas de Impacto

Observemos que as colisões ocorrem entre t_0 e $t_0 + \Delta t$ logo, tem-se a garantia de que não ocorrem interpenetrações em t_0 . O movimento dos pontos em uma zona de impacto, de certo modo, torna-se difícil devido às múltiplas colisões, já que os pontos estão todos em contato uns com os outros.

A idéia então é admitir que esses pontos se movem como um todo permanecendo fixos com respeito uns aos outros, de modo que não ocorram novas colisões dentro de uma zona. Esta hipótese de deslocamento equivale a supor que, dentro de uma zona, a resposta à colisão consiste em *impacto perfeitamente inelástico e contato não deslizando*.

As zonas de impacto se comportarão como objetos rígidos durante o intervalo considerado sendo caracterizadas por uma velocidade de grupo \mathbf{V}_G e uma velocidade angular de grupo $\boldsymbol{\Omega}_G$. \mathbf{V}_G é calculada como a velocidade média⁵ dos n pontos da zona \mathcal{Z} e $\boldsymbol{\Omega}_G$ é calculada com referência ao centro geométrico \mathbf{G} da zona \mathcal{Z} , ou seja

$$\mathbf{V}_G = \frac{1}{n} \sum_{M \in \mathcal{Z}} \mathbf{V}_M \quad \text{e} \quad \boldsymbol{\Omega}_G = \frac{1}{n} \sum_{M \in \mathcal{Z}} \frac{\overrightarrow{\mathbf{GM}} \times (\mathbf{V}_M - \mathbf{V}_G)}{|\overrightarrow{\mathbf{GM}}|^2},$$

onde $\mathbf{G} = \frac{1}{n} \sum_{M \in \mathcal{Z}} \mathbf{M}$.

A resposta à colisão, para todo $M \in \mathcal{Z}$, será dada pela nova velocidade

$$\mathbf{V}'_M = \mathbf{V}_G + \boldsymbol{\Omega}_G \times \overrightarrow{\mathbf{GM}}.$$

5.4.3 Iteração

A hipótese acima garante que não ocorrerá colisões em cada zona durante o intervalo $[t_0, t_0 + \Delta t]$ mas nada garante que o deslocamento das zonas não criem colisões em suas bordas. Para que isso não ocorra, o procedimento iterativo da circunscrição, descrito na Seção 5.4.1, deve ser acoplado ao da resposta específica nas zonas. Assim, o algoritmo pode ser dividido em três fases:

1. detecção e resposta às colisões sem levar em conta as zonas de impacto;
2. nova detecção de colisões memorizando zonas de impacto, se novas colisões aparecerem. Aplica-se a resposta específica à colisão nas zonas detectadas;

⁵Por simplicidade, supomos que todos os pontos têm mesma massa. Não sendo esse o caso, as velocidades seriam calculadas com relação ao centro de massa.

3. iteração da segunda fase, crescendo e juntando zonas se necessário, até que não se detecte novas colisões.

Note que este método iterativo ocorre em cada passo de tempo. Apenas quando as zonas são circunscritas e a resposta às colisões resolvidas, a memória dessas zonas é apagada e o algoritmo passa ao próximo passo de tempo.

5.5 Resultados e Discussões

Para a aplicação desse algoritmo, simulamos diversas situações de colisões de tecidos com alguns objetos. Usamos malhas com números diferentes de pontos, objetos com coeficientes de atrito de superfície variados e tecidos com parâmetros de densidade de massa, elasticidade e atrito diversos. A Figura 5.8 mostra a colisão de um tecido de malha 17×17 pontos com uma mesa quadrada (a) e uma retangular (b). As distribuições dos tecidos sobre esses objetos são tais que não há o deslizamento. Na Figura 5.9 temos a colisão, com as mesmas condições e parâmetros do exemplo anterior, do tecido de malha 33×33 com uma mesa redonda. O Apêndice C exibe alguns quadros da animação de Figura 5.8(a).

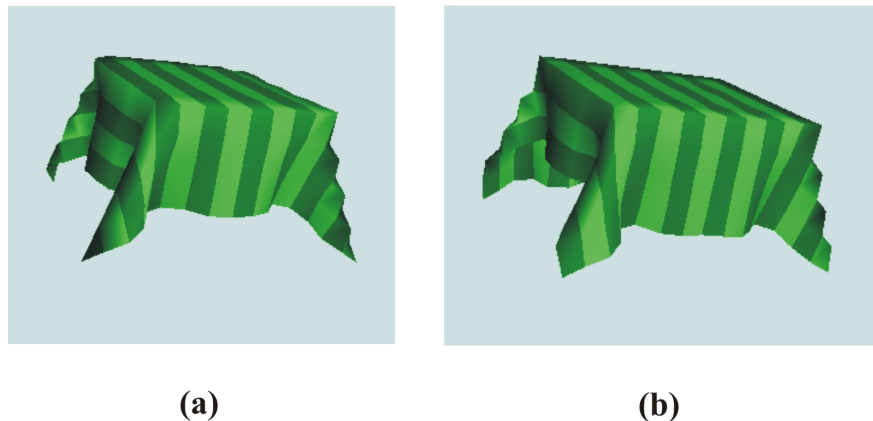


Figura 5.8: Tecido caído sobre mesa quadrada (a) e retangular (b).

Simulamos, também, colisões com objetos não planos. A Figura 5.10 mostra a colisão de tecidos com esferas de diferentes raios. Na parte (a) temos um tecido de malha 17×17 caído sobre a esfera menor e na parte (b) um tecido de malha 33×33 sobre uma esfera maior. Novamente, a distribuição usada é tal que não ocorre o deslizamento do tecido.

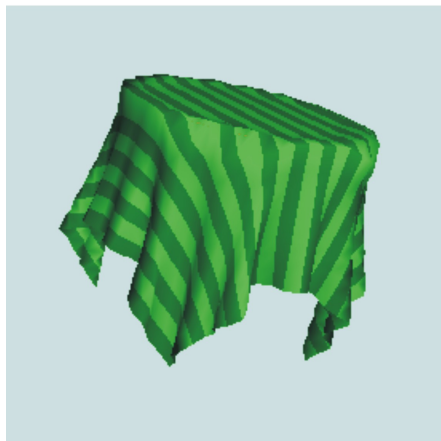


Figura 5.9: Colisão entre o tecido e uma mesa redonda.

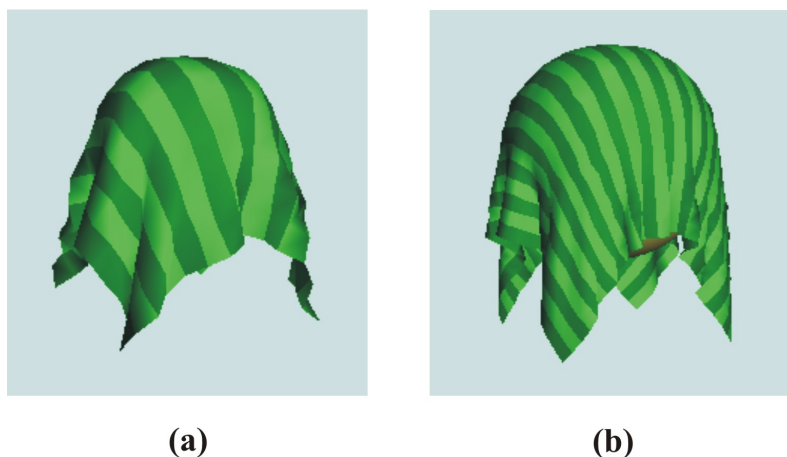


Figura 5.10: Tecido caído sobre esferas de raios diferentes.

Colisões com objetos de baixo coeficiente de atrito são exibidas na Figura 5.11. Um tecido de malha 17×17 desliza sobre um plano inclinado (a) e um tecido de malha 33×33 sobre uma esfera (b). São mostrados, para cada objeto de colisão, dois instantes da simulação. O Apêndice C mostra alguns quadros da animação da Figura 5.11(a).

Embora em nosso trabalho não enfoque a eficiência computacional dos modelos e algoritmos de colisões, cabe, ao menos no caso dos algoritmos de colisões, um comentário a respeito do tempo gasto para as simulações. Fixado o número de pontos da malha dos objetos em 33×33 , obtivemos, para duas malhas de pontos do tecido diferentes, os dados:

- malha de 17×17 : entre 30s e 3min de processamento para 2s de simulação;
- malha de 33×33 : entre 2min e 8min de processamento para 2s de simulação.

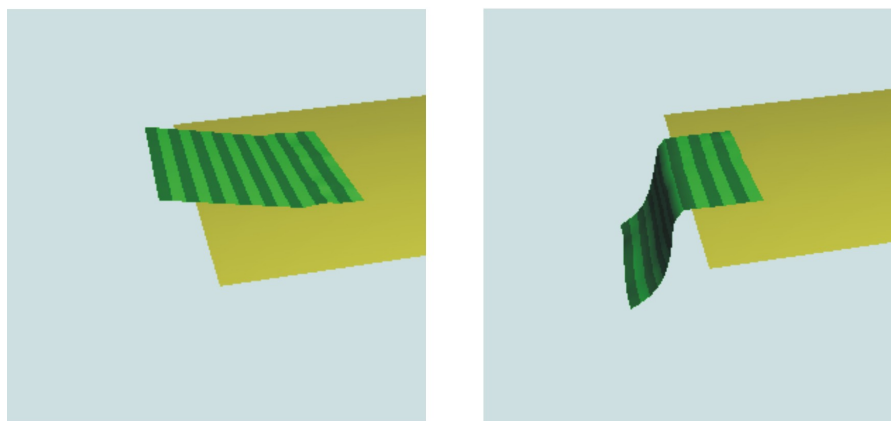
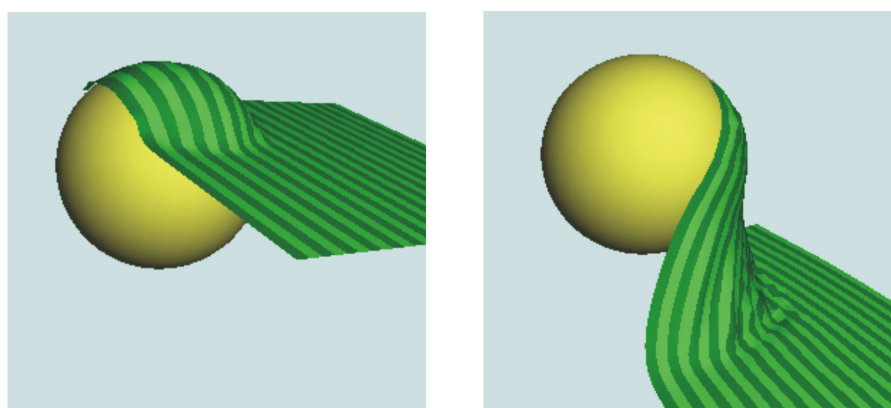
**(a) $t = 0.5s$ e $t = 1.2s$** **(b) $t = 0.5s$ e $t = 1.8s$**

Figura 5.11: Tecido deslizando sobre um plano inclinado (a) e uma esfera (b).

Esses tempos também dependem da área do tecido em contato com o objeto e do tempo desse contato. De fato, maiores áreas de contato e tempo de colisão significam mais colisões a serem tratadas, portanto mais tempo de processamento. O Apêndice C traz, para os leitores curiosos, os tempos de processamento de alguns exemplos expostos em nosso trabalho.

O tempo de Processamento aumenta exponencialmente com a quantidade de pontos da malha do tecido, assim, para diminuí-lo, deve-se usar o menor número de pontos possível nas simulações. Mas isso tem suas desvantagens. O tecido perde realismo na sua forma (Figura 5.10) e apresenta problemas na colisão, principalmente com bordas e quinas. Note, na Figura 5.9, que as dobras do tecido nas bordas da mesa redonda não acompanham o formato real deste objeto.

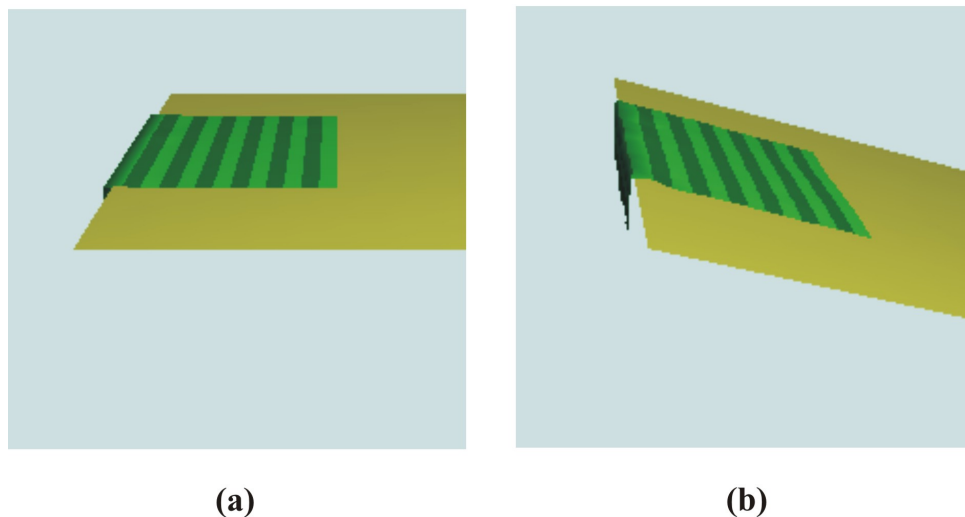


Figura 5.12: O tecido fica “preso” à borda de um plano quando este é inclinado.

Ainda falando em objetos com bordas e quinas, temos mais uma manifestação de falta de realismo associada a eles: o tecido não desliza corretamente na superfície desses objetos. Na Figura 5.12(a) temos um tecido assentado e imóvel sobre uma mesa horizontal. Num determinado instante, inclinamos a mesa mas o tecido não desliza; fica “preso” à borda (Figura 5.12(b)).

Apesar de Provot usar, em seu algoritmo, a triangulação da malha do tecido por retângulos cortados pela mesma diagonal (induzimos isso pelos diagramas e figuras apresentadas em seu artigo [38]), usamos, em nossa implementação, o mesmo estilo de triangulação usado na malha do modelo de Bridson et al. [11] (descrita na Seção 4.1).

Algoritmo de Colisão Para o Modelo de Malha Triangular

Apresentamos, nesse capítulo, o algoritmo de colisões proposto por Bridson et al. [10] aplicado ao modelo de malha triangular [11] outrora estudado. Esse algoritmo foi concebido em data anterior ao modelo deformável em que o utilizaremos, mas seu funcionamento independe do modelo de partículas utilizado.

O algoritmo assemelha-se ao estudado para malha retangular (Capítulo 5) na detecção de colisões e no tratamento para múltiplas colisões. Entretanto, este considera a espessura do tecido e erros de arredondamento; detalhes negligenciados pelo algoritmo anterior. Também aplica forças de repulsão, ao detectar proximidade de colisão, diminuindo a carga para a detecção de colisões (evitando que boa parte delas ocorram) e aumentando o realismo das simulações. Outro ponto forte desse algoritmo está numa aplicação mais coerente da resposta à colisão e proximidade sobre os vértices envolvidos.

Situações diversas são apresentadas como exemplos do funcionamento do algoritmo, acompanhadas de comentários pertinentes sobre a suas características e sua implementação.

6.1 Detecção de Proximidade e Colisão

Para acelerar o processo de detecção de proximidade (para o cálculo da repulsão) e de interseção (para o cálculo da colisão), o algoritmo usa a hierarquia de volumes envolventes baseada em AABB (Seção 5.2.1).

Para o cálculo da repulsão, a cada passo de tempo, calcula-se uma caixa envolvendo cada triângulo aumentada pela espessura do tecido para a construção da hierarquia. Em nosso caso, usamos 1mm de espessura. Para o cálculo da colisão, recalcula-se a hierarquia tomando caixas envolvendo os triângulos em t_0 e $t_0 + \Delta t$ (exatamente como descrito na Seção 5.2.1) aumentadas por uma tolerância à erros de arredondamento da máquina. Em nossa implementação atribuímos 10^{-6} a essa tolerância. Em ambos os casos, cálculo da repulsão e cálculo da colisão, o algoritmo percorre recursivamente as árvores testando interseções entre as caixas e, caso haja interseção, testará proximidades e colisões entre pontos e triângulos e entre pares de arestas contidas nessas caixas.

Diremos que um ponto está *próximo* de um triângulo se estiver a uma distância (do triângulo) menor que a espessura do tecido. Um ponto colidirá com um triângulo se estiver a uma distância menor que a margem de tolerância à erros. Daqui em diante usaremos a notação \mathbf{x}_{ij} representando $\mathbf{x}_i - \mathbf{x}_j$.

Para saber se um ponto \mathbf{x}_4 está próximo a um triângulo $\mathbf{x}_1\mathbf{x}_2\mathbf{x}_3$ de normal \mathbf{n} testa-se primeiro se o ponto está próximo ao plano que contém o triângulo, ou seja, se $|\mathbf{x}_{43} \cdot \mathbf{n}| < h$, onde h é a espessura do tecido. Caso isso ocorra, projetamos o ponto no plano calculando suas coordenadas baricêntricas w_1, w_2 e w_3 com respeito ao triângulo:

$$\begin{bmatrix} \mathbf{x}_{13} \cdot \mathbf{x}_{13} & \mathbf{x}_{13} \cdot \mathbf{x}_{23} \\ \mathbf{x}_{13} \cdot \mathbf{x}_{23} & \mathbf{x}_{23} \cdot \mathbf{x}_{23} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{13} \cdot \mathbf{x}_{43} \\ \mathbf{x}_{23} \cdot \mathbf{x}_{43} \end{bmatrix}$$

$$w_1 + w_2 + w_3 = 1.$$

Se essas coordenadas estiverem no intervalo $[-\delta, 1 + \delta]$, onde δ é h dividido pelo comprimento característico¹ do triângulo, registramos a proximidade.

Para saber se uma aresta $\mathbf{x}_1\mathbf{x}_2$ está próxima a outra aresta $\mathbf{x}_3\mathbf{x}_4$ procuramos um par de pontos, um em cada aresta, que estejam mais próximos entre si e medimos a distância entre eles. Descartando-se o caso degenerado $|\mathbf{x}_{21} \times \mathbf{x}_{43}|$ menor que a margem de tolerância à erros, isso é feito resolvendo-se a equação normal

$$\begin{bmatrix} \mathbf{x}_{21} \cdot \mathbf{x}_{21} & -\mathbf{x}_{21} \cdot \mathbf{x}_{43} \\ -\mathbf{x}_{21} \cdot \mathbf{x}_{43} & \mathbf{x}_{43} \cdot \mathbf{x}_{43} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{21} \cdot \mathbf{x}_{31} \\ -\mathbf{x}_{43} \cdot \mathbf{x}_{31} \end{bmatrix}.$$

¹Média aritmética dos comprimentos das arestas.

Se os pontos estão sobre as arestas, isto é, $0 \leq a, b \leq 1$, está terminado. Caso contrário, os “puxamos” para as extremidades mais próximas de suas respectivas arestas (no caso de um ponto pertencer a primeira aresta e o outro não pertencer à segunda, puxamos apenas o que não pertence obtendo a primeira parte da resposta). O ponto que tiver maior deslocamento é a primeira parte da resposta, com a segunda parte sendo a projeção do primeiro ponto sobre a reta suporte da segunda aresta. Deve-se “puxar” esse novo ponto para a extremidade mais próxima da aresta, caso ele não pertença a ela. Salvamos a direção apontando de um ponto a outro como a normal e os pesos $0 \leq a, b \leq 1$, de modo que temos as posições dos pontos ao longo das arestas, a saber $\mathbf{x}_1 + a\mathbf{x}_{21}$ e $\mathbf{x}_3 + b\mathbf{x}_{43}$.

Para detectar uma colisão entre um ponto e um triângulo ou entre arestas, é necessário observar quando os quatro pontos estão em um mesmo plano (como descrito no algoritmo de Provot [38] nas Seções 5.1.1 e 5.1.2). Isso nos leva à equação cúbica

$$(\mathbf{x}_{21} + t\mathbf{v}_{21}) \times (\mathbf{x}_{31} + t\mathbf{v}_{31}) \cdot (\mathbf{x}_{41} + t\mathbf{v}_{41}) = 0,$$

donde injetamos as raízes, descartando-se aquelas que estão fora do intervalo $[0, \Delta t]$, em ordem crescente, nos testes de proximidade já descritos. Caso os elementos estejam dentro da proximidade de colisão (nossa tolerância à erros, 10^{-6}), registramos a colisão.

6.2 Resposta ao Contato e à Colisão

Precisamos lidar com pontos do tecido interiores a triângulos e arestas computando suas velocidades ou aplicando impulsos a eles. O algoritmo determina as velocidades desses pontos e os impulsos, necessários à correção nas velocidades dos vértices envolvidos, através de interpolação linear.

6.2.1 Determinando Velocidades e a Aplicação de Impulsos

Se um ponto interior a um triângulo $\mathbf{x}_1\mathbf{x}_2\mathbf{x}_3$ possui coordenadas baricêntricas w_1, w_2 e w_3 (calculadas durante a detecção de proximidade ou colisão) sua velocidade interpolada será $w_1\mathbf{v}_1 + w_2\mathbf{v}_2 + w_3\mathbf{v}_3$ e, de modo análogo, um ponto interior de posição relativa a (distância do ponto à extremidade \mathbf{x}_2) a uma aresta $\mathbf{x}_1\mathbf{x}_2$ possui velocidade interpolada $(1 - a)\mathbf{v}_1 + a\mathbf{v}_2$.

Se um impulso de magnitude I na direção \mathbf{n} precisa ser aplicado a dois pontos do tecido, $I\mathbf{n}$ ao primeiro e $-I\mathbf{n}$ ao segundo, devemos aplicá-lo aos vértices do triângulo, tendo por pesos as coordenadas baricêntricas, de modo que a mudança na velocidade relativa dos pontos seja alcançada. Quando um ponto interior ao triângulo $\mathbf{x}_1\mathbf{x}_2\mathbf{x}_3$ com coordenadas baricêntricas w_1, w_2 e w_3 interage com o ponto \mathbf{x}_4 , calculamos o impulso ajustado

$$\tilde{I} = \frac{2I}{1 + w_1^2 + w_2^2 + w_3^2}$$

e as novas velocidades

$$\begin{aligned} \mathbf{v}'_i &= \mathbf{v}_i + w_i \frac{\tilde{I}}{m} \mathbf{n} \quad i = 1, 2, 3 \\ \mathbf{v}'_4 &= \mathbf{v}_4 - \frac{\tilde{I}}{m} \mathbf{n}, \end{aligned}$$

considerando que todos os vértices tenham mesma massa. Para o caso aresta-aresta onde um ponto de posição relativa a (distância do ponto à extremidade \mathbf{x}_2) ao longo da aresta $\mathbf{x}_1\mathbf{x}_2$ interage com outro de posição relativa b (distância do ponto à extremidade \mathbf{x}_4) ao longo da aresta $\mathbf{x}_3\mathbf{x}_4$, calculamos o impulso ajustado

$$\tilde{I} = \frac{2I}{a^2 + (1-a)^2 + b^2 + (1-b)^2}$$

e as novas velocidades

$$\begin{aligned} \mathbf{v}'_1 &= \mathbf{v}_1 + (1-a) \frac{\tilde{I}}{m} \mathbf{n} & \mathbf{v}'_2 &= \mathbf{v}_2 + a \frac{\tilde{I}}{m} \mathbf{n} \\ \mathbf{v}'_3 &= \mathbf{v}_3 - (1-b) \frac{\tilde{I}}{m} \mathbf{n} & \mathbf{v}'_4 &= \mathbf{v}_4 - b \frac{\tilde{I}}{m} \mathbf{n}. \end{aligned}$$

Aplicando os impulsos dessa forma são gerados os torques apropriados para interações com pontos localizados fora do centro de massa dos triângulos e arestas.

6.2.2 Repulsão

As forças de repulsão têm por finalidade reduzir o número de colisões (processo mais caro computacionalmente) tornando o algoritmo mais eficiente. Como o tecido possui uma espessura fixa, as forças de repulsão são aplicadas entre regiões do tecido que tenham essa proximidade. Isso permite que o tecido deslize sobre si ou sobre objetos sem obstáculos excessivos causados por erros de discretização resultantes da representação de superfícies por triângulos.

Se um par, ponto-triângulo ou aresta-aresta, estiver próximo, dois tipos de força de repulsão são aplicadas. A primeira baseada em uma colisão inelástica e a segunda baseada em uma mola.

Suponha que dois pontos do tecido, um dentro de um triângulo e outro vértice da malha ou ambos interiores a duas arestas, estejam próximos com velocidade relativa V_N menor que zero (aproximando-se), na direção normal (Veja a Seção 6.2.1 para detalhes da determinação de velocidades dos pontos interiores e a Seção 6.1 para detalhes sobre a normal usada nas interações ponto-triângulo e aresta-aresta). Para eliminar a colisão iminente, aplica-se um impulso inelástico de magnitude $mV_N/2$ na direção normal (veja na Seção 6.2.1 como distribuir esse impulso nos vértices envolvidos).

Quando regiões do tecido estão muito próximas, ocorre a compressão de suas fibras. Assim, a segunda força é usada para modelar essa compressão. Essa força deve ser proporcional à sobreposição, entre as regiões, excedente à espessura do tecido. Como o algoritmo evitará qualquer tipo de interseção, esse tipo de força de repulsão está limitada a um máximo quando os objetos se tocam. Além disso, o algoritmo limita essa força de modo que os objetos nunca sejam distanciados mais que a zona de sobreposição num único passo de tempo. Isso é importante pois a magnitude da força de repulsão será usada para a modelagem da fricção (Seção 6.2.3).

Na modelagem da força de repulsão por mola usa-se a seguinte expressão para medir a sobreposição (Figura 6.1)

$$d = h - (\mathbf{x}_4 - w_1\mathbf{x}_1 - w_2\mathbf{x}_2 - w_3\mathbf{x}_3) \cdot \mathbf{n},$$

onde obtemos a força kd na direção normal, sendo h a espessura do tecido, k a constante da mola e $\Delta t kd$ o impulso. Como dito anteriormente, para limitar o distanciamento causado pela força, reduziremos a sobreposição em, no máximo, uma fração (por exemplo, 10%) por passo de tempo. Assim, se a componente normal da velocidade relativa $V_N \geq 0.1d/\Delta t$ a repulsão não é aplicada. Caso contrário, calcula-se o impulso de magnitude

$$I = -\min\left(\Delta t kd, m\left(\frac{0.1d}{\Delta t} - V_N\right)\right)$$

e distribui-se pelos vértices envolvidos como dito na Seção 6.2.1.

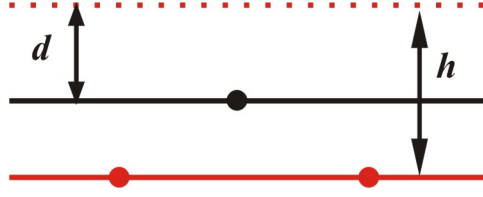


Figura 6.1: A medida da sobreposição entre regiões do tecido: d representa a sobreposição e h a espessura do tecido.

6.2.3 Fricção

A força de repulsão da Seção 6.2.2 é a negativa da força normal F_N que pressiona o tecido. Portanto, a força de fricção, que possui mesma direção da velocidade relativa tangencial pré-fricção \mathbf{V}_T^{pre} mas de sentido oposto ao dela, tem magnitude máxima μF_N , onde μ é o coeficiente de atrito. Isso determina um impulso de magnitude $\mu F_N \Delta t$ e uma mudança na velocidade relativa tangencial de, no máximo, $\mu F_N \Delta t / m$ (assumindo massa igual para todas as partículas). O cálculo da nova velocidade pode ser simplificado observando-se que $F_N \Delta t / m$ é a variação da velocidade relativa ΔV_N , na direção normal, que pode ser diretamente calculada no algoritmo de repulsão. Assim, o decréscimo na magnitude da velocidade relativa tangencial será $\min(\mu \Delta V_N, |\mathbf{V}_T^{pre}|)$, isto é, a velocidade relativa tangencial final é

$$\mathbf{V}_T = \max\left(1 - \mu \frac{\Delta V_N}{|\mathbf{V}_T^{pre}|}, 0\right) \mathbf{V}_T^{pre}.$$

Finalmente, aplicamos o impulso de magnitude

$$I = m \frac{\left(1 - \max\left(1 - \mu \frac{\Delta V_N}{|\mathbf{V}_T^{pre}|}, 0\right)\right) |\mathbf{V}_T^{pre}|}{2}$$

na direção de \mathbf{V}_T^{pre} , para as proximidades ponto-triângulo e aresta-aresta.

6.2.4 Colisão

As forças de repulsão não evitam totalmente as interpenetrações porque testam posições em momentos discretos do tempo. Para um tratamento robusto de colisões, deve-se considerar o “caminho” do tecido entre passos de tempo. Por isso tomamos, na Seção 6.1, caixas envolvendo o objeto e sua futura posição.

Quando uma colisão é registrada (Seção 6.1), o algoritmo leva em conta se os objetos estão se aproximando ou se afastando. Se estão se aproximando, um impulso

de repulsão completamente inelástico é aplicado. Caso estejam se afastando, é aplicado um impulso de repulsão baseado numa mola. Em ambos os casos, esse procedimento é análogo ao descrito na Seção 6.2.2.

6.2.5 Zonas de Impacto

O algoritmo utiliza a técnica proposta por Provot [38] (já descrita na Seção 5.4.1 do Capítulo 5) para o tratamento de múltiplas colisões. Entretanto, Bridson et al. [10] afirmam que a expressão para a velocidade angular do trabalho de Provot está incorreta e apresentam as seguintes correções.

Primeiro calcula-se o centro de massa e a velocidade média da zona de impacto

$$\mathbf{x}_{CM} = \frac{\sum_i m \mathbf{x}_i^n}{\sum_i m}, \quad \mathbf{v}_{CM} = \frac{\sum_i m \mathbf{v}_i^{n+1/2}}{\sum_i m}$$

e em seguida o momento angular dos vértices com relação ao centro de massa

$$\mathbf{L} = \sum_i m (\mathbf{x}_i^n - \mathbf{x}_{CM}) \times (\mathbf{v}_i^{n+1/2} - \mathbf{v}_{CM})$$

e o tensor (3×3) de inércia da configuração atual dos vértices, δ representando o tensor identidade (veja o Apêndice B),

$$I = \sum_i m (|\mathbf{x}_i^n - \mathbf{x}_{CM}|^2 \delta - (\mathbf{x}_i^n - \mathbf{x}_{CM}) \otimes (\mathbf{x}_i^n - \mathbf{x}_{CM})).$$

A velocidade angular do corpo rígido que preserva o momento angular será $\vec{\omega} = I^{-1} \mathbf{L}$ e a nova velocidade do vértice i será $\mathbf{v}_i^{n+1/2} = \mathbf{v}_{CM} + \vec{\omega} \times (\mathbf{x}_i - \mathbf{x}_{CM})$. Entretanto, procura-se uma velocidade média sobre um intervalo finito Δt tal que as novas posições $\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t \mathbf{v}^{n+1/2}$ correspondam *exatamente* a um movimento rígido. Se essa condição não é cumprida (como acontece em Provot [38]) interpenetrações podem ocorrer. Aceitando pequenos erros ($\mathcal{O}(\Delta t)$) no eixo e no ângulo de rotação, o algoritmo considera que a velocidade $\vec{\omega}$ permaneça constante ao longo do passo de tempo. Calculando as componentes fixa e de rotação da posição

$$\mathbf{x}_F = \left[(\mathbf{x}_i - \mathbf{x}_{CM}) \cdot \frac{\vec{\omega}}{|\vec{\omega}|} \right] \frac{\vec{\omega}}{|\vec{\omega}|}, \quad \mathbf{x}_R = \mathbf{x}_i - \mathbf{x}_{CM} - \mathbf{x}_F$$

obtém-se a posição final

$$\mathbf{x}_i^{n+1} = \mathbf{x}_{CM} + \Delta t \mathbf{v}_{CM} + \mathbf{x}_F + \cos(\Delta t |\vec{\omega}|) \mathbf{x}_R + \text{sen}(\Delta t |\vec{\omega}|) \left(\frac{\vec{\omega}}{|\vec{\omega}|} \times \mathbf{x}_R \right)$$

e, portanto, a nova velocidade média será

$$\mathbf{v}_i^{n+1/2} = \frac{(\mathbf{x}_i^{n+1} - \mathbf{x}_i^n)}{\Delta t}.$$

6.3 Aplicação do Algoritmo

O processamento de colisões realizado pelo algoritmo é totalmente separado da evolução interna do modelo deformável (e de todos os objetos em cena). Isso permite o seu uso com qualquer modelo de partículas.

A partir de um tempo $t = 0$ com posições \mathbf{x}^0 e velocidades \mathbf{v}^0 o algoritmo é aplicado como segue, para $n = 0, 1, 2, \dots$:

- Selecionar um passo de tempo Δt e fazer $t^{n+1} = t^n + \Delta t$;
- Avançar para as velocidades e posições candidatas $\bar{\mathbf{v}}^{n+1}$ e $\bar{\mathbf{x}}^{n+1}$ com a dinâmica interna do modelo deformável;
- Calcular a velocidade média $\bar{\mathbf{v}}^{n+1/2} = (\bar{\mathbf{x}}^{n+1} - \mathbf{x}^n)/\Delta t$;
- Testar proximidade com \mathbf{x}^n , aplicando forças de repulsão e fricção à velocidade média, obtendo $\tilde{\mathbf{v}}^{n+1/2}$;
- Testar as trajetórias lineares de \mathbf{x}^n com $\tilde{\mathbf{v}}^{n+1/2}$ para colisões obtendo a velocidade média final $\mathbf{v}^{n+1/2}$;
- Calcular a posição final $\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t \mathbf{v}^{n+1/2}$;
- Caso não hajam repulsões ou colisões, fazer $\mathbf{v}^{n+1} = \bar{\mathbf{v}}^{n+1}$. Caso contrário, avançar (implicitamente) a velocidade média $\mathbf{v}^{n+1/2}$ para \mathbf{v}^{n+1} , ou seja, deve-se resolver a equação $\mathbf{v}^{n+1} = \mathbf{v}^{n+1/2} + \frac{\Delta t}{2} \mathbf{a}(t^{n+1}, \mathbf{x}^{n+1}, \mathbf{v}^{n+1})$.

Em nossa implementação, tomamos a aproximação $\mathbf{v}^{n+1} = \mathbf{v}^{n+1/2}$, ou seja, suprimimos o último passo da lista anterior. No modelo deformável de Bridson et al. [11], onde aplicamos esse algoritmo, essa última etapa representa a atuação de forças dependentes da velocidade (veja a Seção 4.2). Para pontos sob efeito do tratamento de repulsão ou colisão, esse efeito é pouco notado (como ocorre em nossas simulações, Seção 6.4). Assim, evitamos o cômputo dessa etapa implícita. No modelo de

malha retangular de Provot [37], o último ítem do algoritmo de colisões é realmente desnecessário já que nesse modelo as velocidades \mathbf{v}^{n+1} e $\mathbf{v}^{n+1/2}$ coincidem (Seção 3.1).

6.4 Resultados e Discussões

Como temos feito ao longo de nosso estudo, apresentamos simulações diversas para a observação do funcionamento dos algoritmos e, em particular, para comparar seus desempenhos em situações semelhantes. A Figura 6.2 mostra tecidos de malha 17×17 pontos colidindo com uma mesa quadrada (a) e uma mesa retangular (b). Um tecido de malha 33×33 pontos colidindo com uma mesa redonda é mostrado na Figura 6.3. Nesses exemplos, as distribuições dos tecidos sobre as mesas são tais que eles não deslizam. O Apêndice C exibe alguns quadros da animação de Figura 6.2(a).

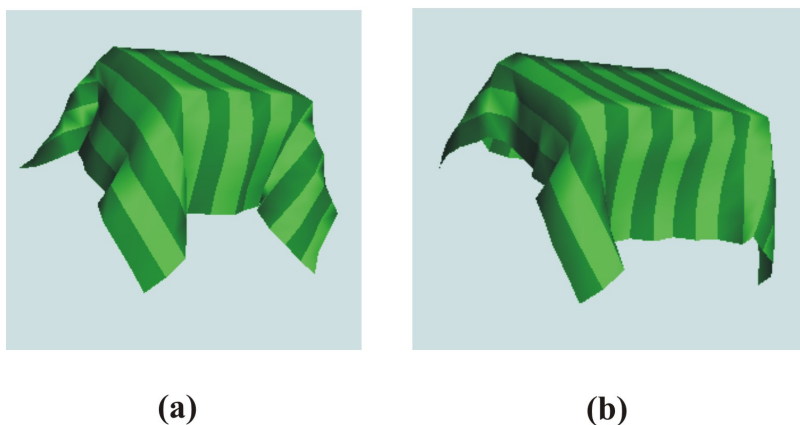
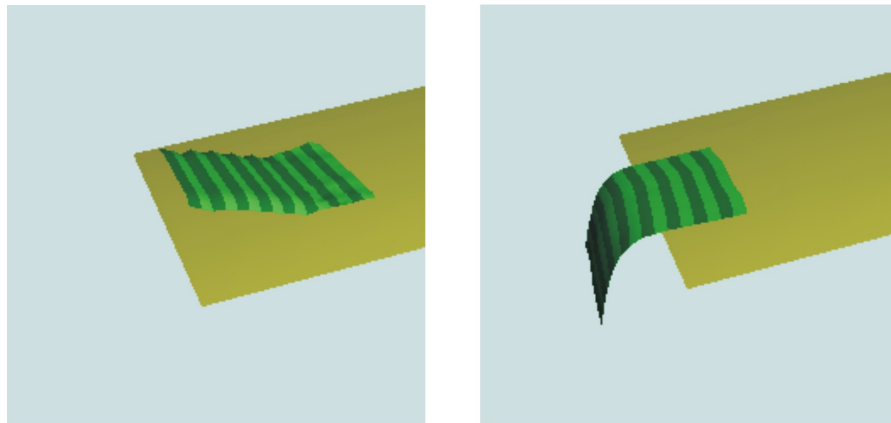


Figura 6.2: Tecido caído sobre mesa quadrada (a) e retangular (b).

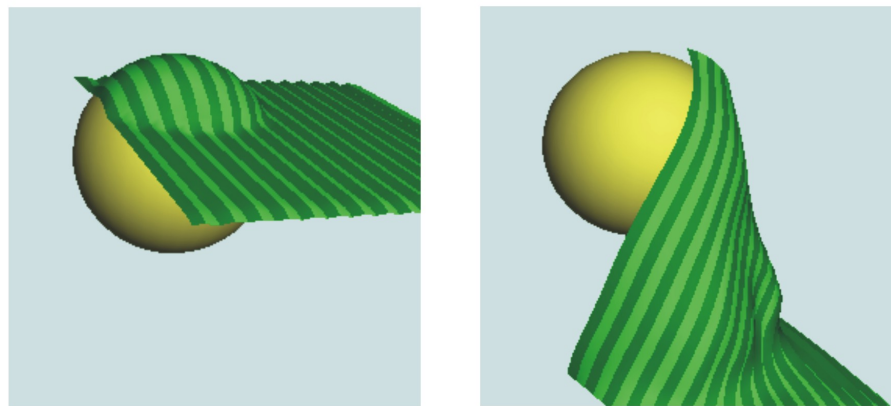


Figura 6.3: Colisão entre o tecido e uma mesa redonda.

Semelhante ao que fizemos na Seção 5.5 para o algoritmo estudado naquele capítulo, simulações de colisões com objetos de baixo coeficiente de atrito são exibidas na Figura 6.4. O um tecido de malha 17×17 desliza sobre um plano inclinado (a) e um tecido de malha 33×33 sobre uma esfera (b). São mostrados, para cada objeto de colisão, dois instantes da simulação. O Apêndice C mostra alguns quadros da animação da Figura 6.4(a).



(a) $t = 0.5s$ e $t = 1.5s$



(b) $t = 0.5s$ e $t = 1.5s$

Figura 6.4: Tecido deslizando sobre um plano inclinado (a) e uma esfera (b).

Desde já, podemos notar características marcantes do algoritmo. Observe, nos exemplos da Figura 6.2, como a parte do tecido que toca as bordas das mesas está inclinada (compare com a Figura 5.8). Isso ocorre porque os impulsos não são calculados e aplicados aos vértices considerando a real posição do ponto de colisão (Seções 6.1 e 6.2.1). Lembramos que, no algoritmo de Provot [38], consideram-se os centros

de massa dos objetos envolvidos como pontos de colisão (Seção 5.3.3). A Figura 6.5 mostra o que ocorre nesse tipo de situação: na parte (a) vemos (numa representação de alguns vértices e arestas, onde a cor verde representa o objeto) o comportamento da malha quando o tecido cai sobre uma mesa usando o algoritmo de Provot [38] e na parte (b) usando o algoritmo de Bridson et al. [10]. Essa forma coerente de aplicação de impulsos, do algoritmo [10], gera o torque responsável pelo movimento da aresta, representado pelas setas vermelhas na figura. Essa característica faz com que o tecido

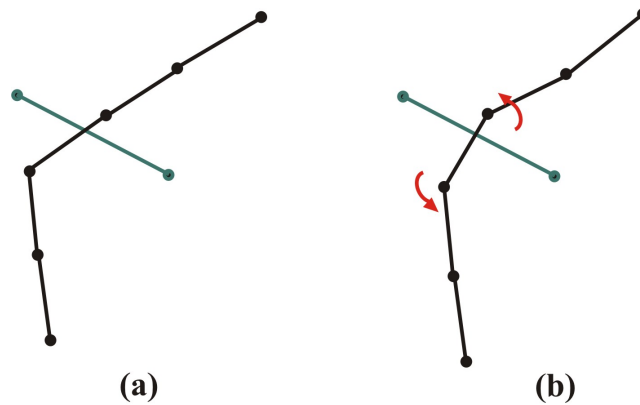


Figura 6.5: Colisão de arestas: (a) com o algoritmo de Provot e (b) com o algoritmo de Bridson et al.

deslize sobre superfícies com quinas. A representação da Figura 6.6(a) mostra que, com o algoritmo [38], o tecido fica preso à quina (veja também a Figura 5.12), enquanto que com o algoritmo [10], na parte (b), o tecido desliza sobre a superfície. A Figura

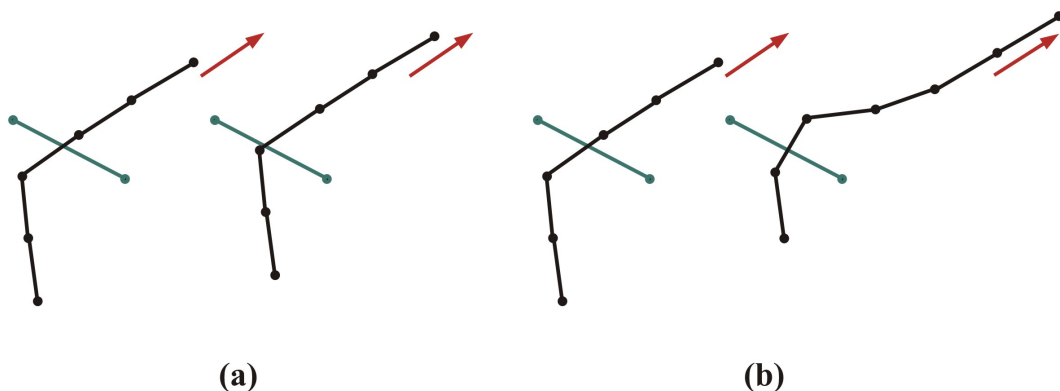


Figura 6.6: Em (a), com o algoritmo [38], não há deslizamento e em (b), com o algoritmo [10], ocorre o deslizamento.

6.7 ilustra o comportamento do tecido (de malha 17×17) sobre um plano inclinado em contraste ao exemplo da Figura 5.12 do capítulo anterior.

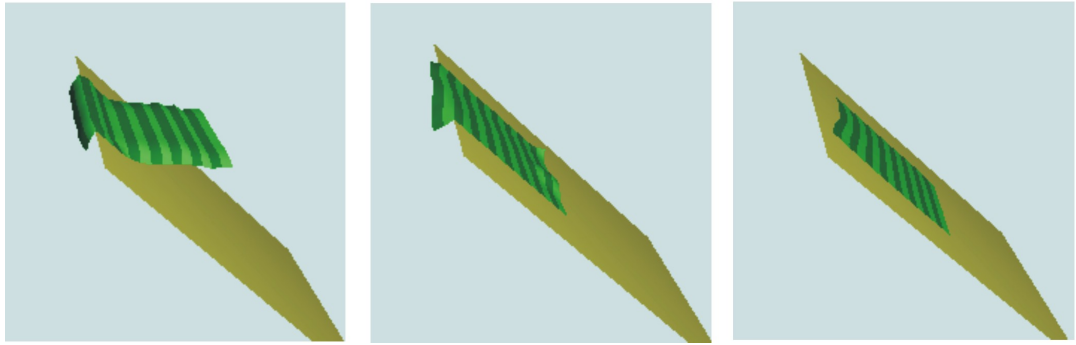


Figura 6.7: O tecido não fica preso à borda do plano.

Na verdade, o algoritmo de colisões de Provot [38] não contempla adequadamente essas situações mais complexas uma vez que foi concebido para o design de roupas, normalmente postas sobre manequins virtuais. Esses manequins possuem formas suaves, sem pontas e/ou quinas. Em todos os exemplos do Capítulo 5, tomamos situações e configurações de parâmetros e posições dos objetos tais que esses problemas não ocorressem ou fossem imperceptíveis. Considerando uma esfera com eixo mais próximo do centro do tecido, comparando com a Figura 5.11(b), observamos que o tecido desliza por um lado ficando preso à esfera pelo outro (Figura 6.8(a)). A ausência de torque nas arestas impede o seu deslizamento. Com objetos em movimento também ocorrerem problemas de mesmas causas. A Figura 6.8(b) mostra o que ocorre a um tecido caído sobre uma esfera, com grande coeficiente de atrito, em movimento. O não tratamento dos erros de arredondamento da máquina ou truncamento de valores provocam interpenetrações e, portanto, perda de realismo.

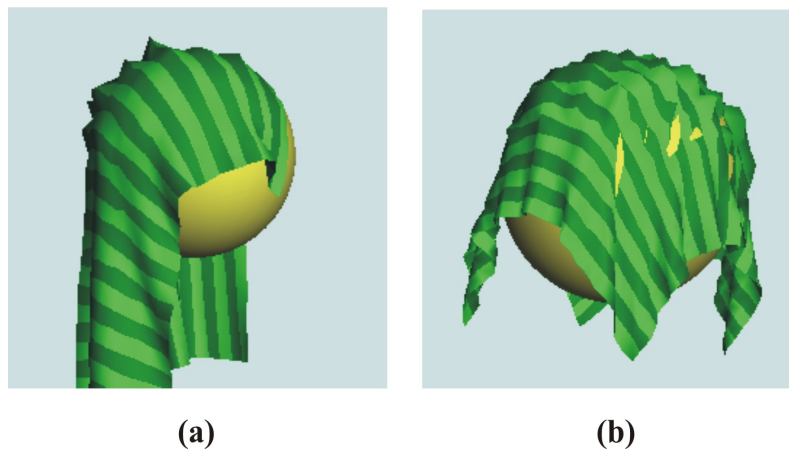


Figura 6.8: Situações-problema para o algoritmo de Provot [38].

Quanto ao tempo de processamento não percebemos grandes diferenças o algoritmo aqui estudado e o do capítulo anterior. Para as mesmas condições e parâmetros escolhidos nas simulações realizadas com o algoritmo de Provot, obtivemos tempos de processamentos muito próximos. A plataforma de desenvolvimento usada para a execução das simulações foi (e sempre será, em todas as simulações) a mesma outrora usada.

As eficientes forças de repulsão e o tratamento robusto de colisões do algoritmo [10], em sinergia, permitem a simulação de movimentos complexos do tecido. O tratamento de auto-colisão com o modelo de fricção produz dobras estáveis, aumentando o realismo. A Figura 6.9 ilustra bem o que foi dito. Nela observamos uma esfera, de malha 65×65 pontos, girando sobre um plano.

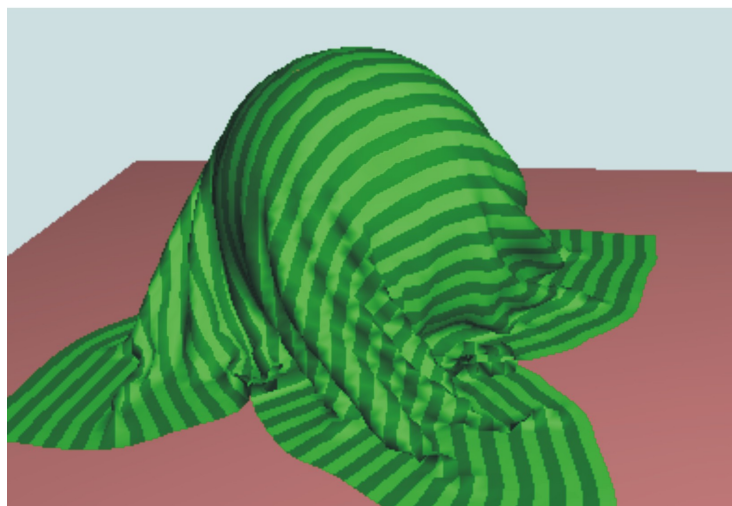


Figura 6.9: Dobras estáveis aumentam o realismo.

A Força de Acoplamento

Neste capítulo propomos o uso de uma força de acoplamento entre deformações tangenciais e normais como solução ao problema de falta de realismo, visto nas Seções 3.4 e 4.5, devido ao não aparecimento espontâneo de dobras e rugas. A técnica pode ser usada para quaisquer dos dois modelos deformáveis estudados, portanto os exemplos aqui presentes podem ser produzidos com ambos.

7.1 O Problema

Retomemos o problema da Seção 3.3: Uma bandeira pendurada com todos os pontos dispostos num plano vertical de mesma direção da gravidade (sem vento). A partir de um instante qualquer “soltamos” um dos pontos fixos. A bandeira cai, ficando presa por um único ponto fixo. A Figura 7.1, mostra o que acontece nessa situação.

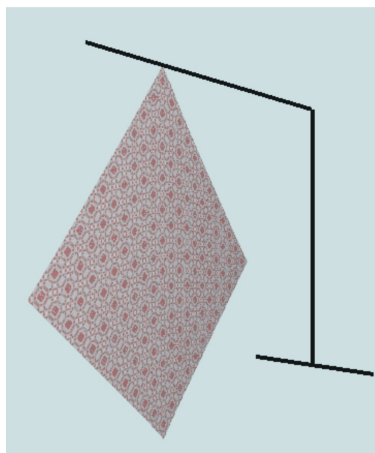


Figura 7.1: As dobras não aparecem espontaneamente.

Isso acontece porque, estando todos os pontos e forças contidos num mesmo plano, não há forças em direções não paralelas ao plano, ou seja, não há componentes de forças na direção normal ao plano. Portanto, o resultado das interações entre os pontos e forças não desloca pontos para fora do plano, assim as rugas não são formadas.

Ocorre que, em geral, os tecidos apresentam uma tendência, quando deformados, de preservação da área o que acarreta o surgimento de forças internas capazes de criar rugas e/ou dobras, conforme a rigidez de curvatura. Este modelo não apresenta a relação entre deformações tangenciais e normais e por isso não responde adequadamente à situação exposta. Na Figura 7.2 mostramos outra situação: num tecido, sem movimento, puxamos duas de suas extremidades não consecutivas. Novamente observamos uma situação não realista.

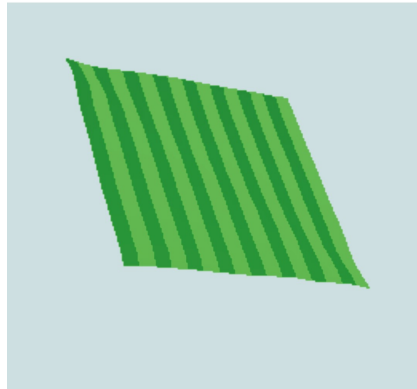


Figura 7.2: Mais uma situação onde rugas não surgem.

7.2 Aplicação da Força de Acoplamento

Nosso procedimento baseia-se na aplicação de uma força proporcional a um coeficiente λ , o *coeficiente de acoplamento*, na direção normal em um ponto quando, no cômputo de uma força elástica estrutural ou de cisalhamento, for detectado uma elongação ou contração da mola. Tomaremos, para a descrição do processo, as molas estruturais. O procedimento age de maneira análoga para as molas de cisalhamento.

Para uma mola estrutural de comprimento natural l_0 temos um comprimento máximo permitido $l_{max} = 1.1l_0$, para o controle da super-elasticidade. O que fazemos é determinar um comprimento crítico l_{crit} para a mola, $l_0 < l_{crit} \leq l_{max}$, a partir do qual a força de acoplamento passará a atuar. Em nossas simulações, utilizamos $l_{crit} = 1.05l_0$.

Quando, durante a avaliação das forças elásticas, o ponto \mathbf{P}_{ij} tiver um vizinho \mathbf{P}_{kl} cuja mola de ligação esteja com tamanho acima de l_{crit} aplicamos ao ponto \mathbf{P}_{ij} a força

$$\mathbf{F}_a = \pm \lambda \Delta_a \mathbf{N}_{ij},$$

onde \mathbf{N}_{ij} é a normal no ponto \mathbf{P}_{ij} , $\Delta_a = l - l_{crit}$ mede o avanço na região a de aplicação (Figura 7.3) e o sinal é escolhido aleatoriamente em cada aplicação da força. Essa escolha aleatória modela o comportamento real dos tecidos quando permite que as rugas geradas variem de posição, ao repetir-se várias vezes uma mesma simulação. Com essa aplicação, as dobras surgem no tecido de acordo com a rigidez de curvatura do mesmo.

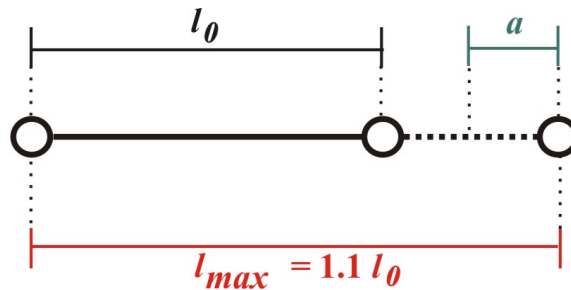


Figura 7.3: Região a de aplicação da força de acoplamento.

Esse procedimento é aplicado da mesma forma para o caso da contração das molas, substituindo-se o comprimento crítico por uma fração menor que o comprimento natural l_0 da mola, por exemplo $l_{crit} = 0.95l_0$. Quando as molas possuírem tamanho inferior a este, a força de acoplamento é aplicada.

A aplicação da força de acoplamento é feita da mesma forma para os modelos deformáveis estudados, Provot [37] e Bridson et al. [11]. Porém, neste último, foi necessária uma pequena alteração, visto que nele não é permitido compressão para as molas (Seção 4.3). Permitimos, exatamente como no modelo de Provot [37], a compressão das molas de forma que podemos aplicar a força de acoplamento nessa situação. Outra particularidade desse mesmo modelo é que ele não possui molas de cisalhamento e sim molas estruturais com comprimentos naturais diferentes (dois em nosso caso). Assim aplicamos a técnica apenas nessas molas usando comprimentos críticos apropriados para cada uma.

A Figura 7.4 mostra o resultado da aplicação das forças de acoplamento para os exemplos da seção anterior. O Apêndice C exhibe alguns quadros da animação da

bandeira pendurada por um ponto com o uso de nosso acoplamento.

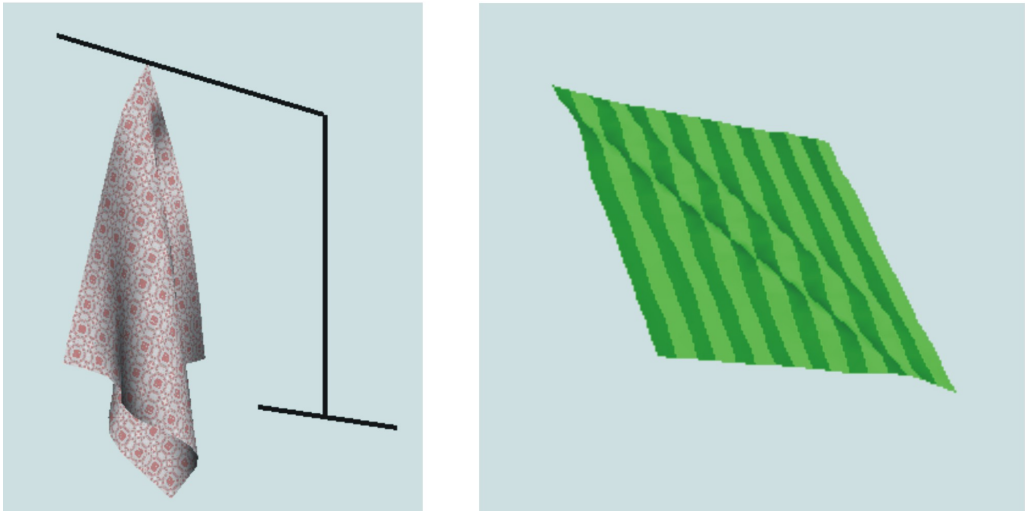


Figura 7.4: Resposta coerente do modelo com forças de acoplamento.

Conclusões e Futuros Trabalhos

Finalizamos, neste capítulo, o nosso estudo sobre modelos deformáveis e algoritmos de colisões com as conclusões e algumas considerações finais. Procuramos, também, traçar metas para a continuidade de nossos estudos sobre os temas através de propostas para futuros trabalhos.

8.1 Conclusões

O modelo de malha retangular de Provot [37] mostrou, para nós, ter maior importância histórica e didática que importância prática. Seu método simples de integração, sua modelagem de curvatura por molas (o que impossibilita a pré-moldagem de dobras) e o seu procedimento de correção da super-elasticidade (que não conserva o momento linear) o tornam um modelo pouco realístico e pouco versátil. Entretanto, nos serviu, adequadamente, como um ponto de partida para nossos estudos.

O modelo de malha triangular de Bridson et al. [11] mostrou-se mais realístico. A malha triangulada representa melhor superfícies curvas, a correção da super-elasticidade com conservação do momento linear traz uma mobilidade comparável à de tecidos reais, o sistema de integração mais elaborado melhora a estabilidade e a modelagem da curvatura por ângulos entre triângulos permitem tanto mais realismo quanto um número maior de aplicações para esse modelo.

Nenhum desses modelos contempla relações entre deformações tangenciais e normais e, por isso, eles não geram dobras e rugas espontaneamente. Propusemos o uso da força de uma acoplamento entre essas deformações, em ambos os modelos, como

solução para esse problema.

Podemos, apropriadamente, repetir a primeira frase dos dois primeiros parágrafos dessa seção para os algoritmos de colisões de Provot [38] e de Bridson et al. [10], respectivamente. O primeiro revelou-se insuficiente para a correta detecção e tratamento de colisões principalmente por negligenciar erros de arredondamento da máquina e erros de truncamento de valores. O segundo considera esses erros, desenvolve uma técnica mais coerente de aplicação de impulsos para a correção das velocidades e corrige o processo de zonas de impacto, proposto pelo algoritmo [38].

8.2 Considerações Finais

Não foi fácil para nós, dentro do curto prazo de um curso de mestrado, estudar e implementar dois modelos deformáveis e dois algoritmos de colisões. Muitos foram os artigos e livros lidos para alcançarmos esse fim. Além disso, tivemos que estudar campos de conhecimentos como os da Física e das Ciências da Computação. Nesse último, estudamos a biblioteca Mesa e o GLUT, programação em C++, estruturas de dados, orientação à objetos (ao menos sobrecarga de operadores), enfim, uma grande quantidade de informação teve de ser digerida.

Artigos remetem à outros artigos em cadeias quase sem fim. Precisamos, então, fazer truncamentos nessas cadeias através de escolhas; estas, quase sempre, direcionadas à simplicidade.

Optamos por não aplicar métodos, como o do gradiente conjugado [39], para a solução de sistemas implícitos. Em vez disso, procuramos soluções alternativas (como em 4.2 e 6.3) que, apesar de simples, servissem aos objetivos iniciais de nosso estudo. Evitamos implementar a etapa de colisões do artigo [11] pois esta necessita de estruturas de dados e técnicas que não caberiam no escopo desse trabalho (essa etapa remete ao estudo de novos conceitos como função distância com sinal, modelos implícitos, entre outros). Também evitamos o pós-processamento por subdivisão no artigo de colisões [10]. Essa técnica é dedicada a uma melhoria na aparência do tecido através da suavização da malha triangular, portanto pôde ser posta à parte não comprometendo nossos objetivos.

Os modelos apresentados funcionaram da forma esperada, mesmo com as restri-

ções feitas nas implementações. Pudemos realizar simulações semelhantes às encontradas nos trabalhos estudados e, além disso, criamos situações novas donde, inclusive, detectamos uma falta de realismo e propomos uma solução.

8.3 Futuros Trabalhos

Uma vez alcançados nossos objetivos iniciais nos estudos de modelos deformáveis, existem várias possibilidades para trabalhos que nos permitam consolidar e expandir nossos conhecimentos na área.

Primeiramente, almejamos concluir a construção dos modelos deformáveis e algoritmos de colisões já estudados, acrescentando o que lhes deixamos faltar ou o que, deliberadamente, alteramos, como dito na seção anterior. Isso implica na tarefa relacionada de desenvolver as estruturas de dados e técnicas complementares necessárias. Procuraremos desenvolver também, nesse ínterim, nossas habilidades para a orientação à objetos, visando as vantagens inerentes a esse tipo de programação.

Para os modelos deformáveis podemos estudar a técnica de correção da superelasticidade (a ordem de ajuste das molas e como o ajuste deve ser feito), a eficiência e estabilidade de seus sistemas de integração numérica e a modelagem mais precisa do atrito e de forças externas (como o vento, por exemplo).

Para o estudo de algoritmos de colisões, devemos submetê-los a um maior número de situações. A construção de manequins móveis virtuais e objetos de geometrias complexas serão passos importantes para o estudo do funcionamento desses algoritmos. Os próprios criadores do algoritmo [10] indicam, para seus futuros trabalhos, o estudo de colisões com objetos pontiagudos e colisões do tecido “engavetado” entre dois objetos. Procuraremos situações como estas em nossos estudos vindouros.

Os custos computacionais também serão objetos de nosso interesse. Para o tratamento de colisões, principalmente, técnicas que garantam rápida convergência em seus resultados contribuem para o baixo custo computacional e maior eficiência do algoritmo.

Pretendemos também ampliar nossos conhecimentos bibliográficos sobre os modelos deformáveis de partículas, algoritmos de colisões, técnicas numéricas, ferramentas computacionais (aplicáveis ao contexto de computação gráfica) e aplicações possíveis

para os modelos (sejam matemáticas, médicas, no entretenimento ou em qualquer outra área de conhecimento).

Apêndice **A**

Métodos Numéricos

Neste apêndice, apresentamos uma breve descrição dos métodos de Euler e trapezoidal, usados em nosso estudo, para a solução numérica de equações diferenciais ordinárias (EDO).

Euler Avançado (FE): o mais simples método para solução de uma EDO. Ele usa a inclinação em cada ponto, calculada com a EDO, para extrapolar e encontrar o próximo ponto:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{x}'_n.$$

Por exemplo, para integrar a EDO $x'(t) = -2x(t) + t$, com $x(0) = 1$, usando o passo $h = 0.1$, fazemos

$$x(0.1) = x(0) + hx'(0) = 1 + (0.1)(-2(1) + 0) = 0.8 .$$

Euler Recuado (BE): esse método, um pouco mais complicado, toma um passo do método FE para usar a inclinação do ponto à frente, e não a do ponto original, para extrapolar a partir do ponto original:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{x}'_{n+1,FE}$$

onde $\mathbf{x}_{n+1,FE}$ é o resultado do passo FE de \mathbf{x}_n e $\mathbf{x}'_{n+1,FE}$ é a inclinação nesse ponto. Por exemplo, para integrar a mesma EDO anterior (com os mesmos parâmetros) usando BE, fazemos

$$x(0.1)_{FE} = x(0) + hx'(0) = 1 + (0.1)(-2(1) + 0) = 0.8$$

e, em seguida,

$$x(0.1) = x(0) + h(x'(0.1)_{FE}) = 1 + (0.1)(-2(0.8) + 0.1) = 0.85 .$$

Método Trapezoidal: este método toma um passo FE para obter $\mathbf{x}_{n+1,FE}$, usa a EDO para o cálculo da inclinação nesse ponto e toma a média entre essa inclinação e a inclinação do ponto atual para o cálculo do próximo ponto:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{2}(\mathbf{x}'_n + \mathbf{x}'_{n+1,FE}).$$

Por exemplo, para integrar a mesma EDO anterior (com os mesmos parâmetros) usando BE, fazemos

$$x(0.1)_{FE} = x(0) + hx'(0) = 1 + (0.1)(-2(1) + 0) = 0.8$$

e, em seguida,

$$x(0.1) = x(0) + \frac{h}{2}(x'(0) + x'(0.1)_{FE}) = 1 + (0.05)[(-2(1) + 0) + (-2(0.8) + 0.1)] = 0.825 .$$

Este último método é o mais preciso dos três, apresentando erros de ordem $\mathcal{O}(h^3)$. Os demais apresentam erros de ordem $\mathcal{O}(h^2)$. Um estudo detalhado sobre erros, estabilidade e convergência dos métodos numéricos para a solução de EDO podem ser encontrados nas referências [16] e [6].

Produto Tensorial e Tensores

Começaremos pela definição de aplicação bilinear e seguiremos com as definições de produto tensorial e tensores. Para nosso estudo, o produto tensorial será tomado apenas sobre dois espaços vetoriais de dimensões finitas, a saber, duas cópias do \mathbb{R}^3 . Para conhecer generalizações do produto tensorial para mais espaços, inclusive de dimensão infinita, além de noções de álgebra de tensores, recomendamos a leitura do livro [31].

Definição 1 *Seja K um corpo e sejam U, V e W espaços vetoriais. A aplicação $\varphi : U \times V \rightarrow W$ é chamada bilinear se for linear em suas duas coordenadas, isto é,*

- $\varphi(u + v, p + q) = \varphi(u, p) + \varphi(v, p) + \varphi(u, q) + \varphi(v, q)$ e
- $\varphi(ku, sp) = ks\varphi(u, p)$,

para quaisquer $u, v \in U$, $p, q \in V$ e $k, s \in K$.

Definição 2 *Sejam U e V espaços vetoriais de dimensões m e n , respectivamente. Um produto tensorial de U por V é um par (\mathcal{Z}, φ) que satisfaz os critérios*

- \mathcal{Z} é um espaço vetorial e $\varphi : U \times V \rightarrow \mathcal{Z}$ é uma aplicação bilinear;
- $\dim(\mathcal{Z}) = mn$ e
- $\varphi(U \times V)$ gera o espaço \mathcal{Z} .

Nomenclatura:

- usamos $U \otimes V$ para o produto tensorial de U por V ;
- representa-se $\varphi(u, v)$ por $u \otimes v$, lendo-se “ u tensor v ”;

- os elementos de $U \otimes V$ são chamados de tensores (ou 2-tensores, pelo uso de dois espaços).

Apresentamos uma (o livro [31] mostra três) construção para o produto tensorial entre dois espaços vetoriais. Sejam $\{e_1, \dots, e_m\}$ e $\{f_1, \dots, f_n\}$ bases dos espaços U e V , respectivamente. Tomemos uma base $\{h_{11}, \dots, h_{ij}, \dots, h_{mn}\}$ de \mathcal{Z} e definamos $\varphi : U \times V \rightarrow \mathcal{Z}$ nos pares (e_i, f_j) por $\varphi(e_i, f_j) = h_{ij}$. Dessa forma, o par (\mathcal{Z}, φ) é um produto tensorial de U por V .

Em nosso caso, identificando \mathbb{R}^9 com o espaço $M_{3 \times 3}$ das matrizes 3×3 tendo $\{e_1, e_2, e_3\}$ base de \mathbb{R}^3 e $\{f_{11}, f_{12}, \dots, f_{32}, f_{33}\}$ base de $M_{3 \times 3}$, definamos $\varphi : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow M_{3 \times 3}$ nos pares (e_i, e_j) por $\varphi(e_i, e_j) = f_{ij}$. Assim, $(M_{3 \times 3}, \varphi)$ é um produto tensorial de \mathbb{R}^3 por \mathbb{R}^3 . Tomando $u = \sum_i \alpha_i e_i$ e $v = \sum_i \beta_i e_i$ em \mathbb{R}^3 teremos então a expressão de $u \otimes v$ dada por

$$u \otimes v = \varphi(u, v) = \begin{pmatrix} \alpha_1 \beta_1 & \alpha_1 \beta_2 & \alpha_1 \beta_3 \\ \alpha_2 \beta_1 & \alpha_2 \beta_2 & \alpha_2 \beta_3 \\ \alpha_3 \beta_1 & \alpha_3 \beta_2 & \alpha_3 \beta_3 \end{pmatrix}.$$

O tensor identidade é representado pela matriz

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Quadros de Animações

O principal objetivo desse apêndice é mostrar quadros de animação de alguns exemplos que constam no trabalho, levando o leitor a ter maior noção do comportamento dinâmico dos modelos deformáveis e algoritmos de colisões.

Incluímos nesse apêndice a descrição das malhas dos objetos e os tempos de animação e de processamento de cada exemplo aqui apresentado, embora, como afirmamos em nossos objetivos, essas observações não façam parte do escopo de nosso trabalho. Apenas queremos, com isso, satisfazer aos espíritos mais curiosos.

Modelo de Malha Retangular: Aqui mostramos (Figura C.1) alguns quadros da animação da bandeira ao vento mostrada na Figura 4.9(a). Com a malha de 13×13 pontos, nesse exemplo, obtivemos um tempo de processamento de 1.2 segundos para 3 segundos de animação.

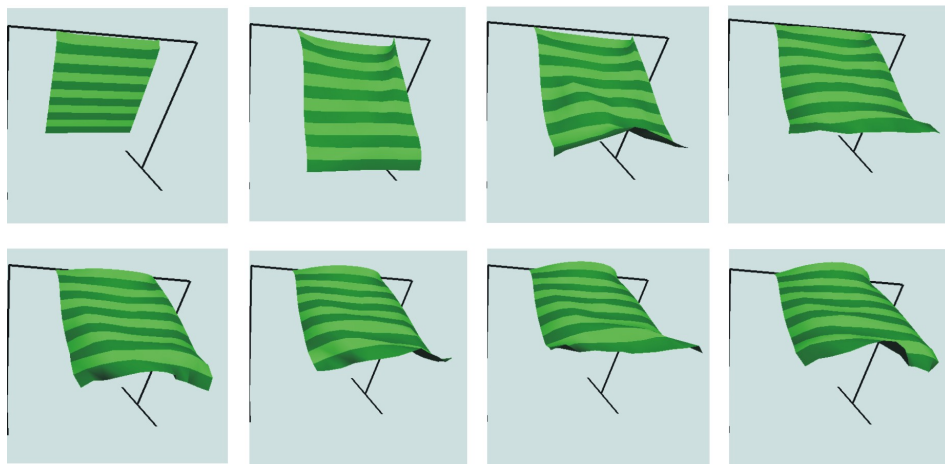


Figura C.1: Vela ao vento com o modelo de malha retangular.

Modelo de Malha Triangular: Na Figura C.2 exibimos alguns quadros da animação da bandeira ao vento mostrada na Figura 4.9(b). Com a malha de 13×13 pontos, nesse exemplo, obtivemos um tempo de processamento de 5 segundos para 3 segundos de animação.

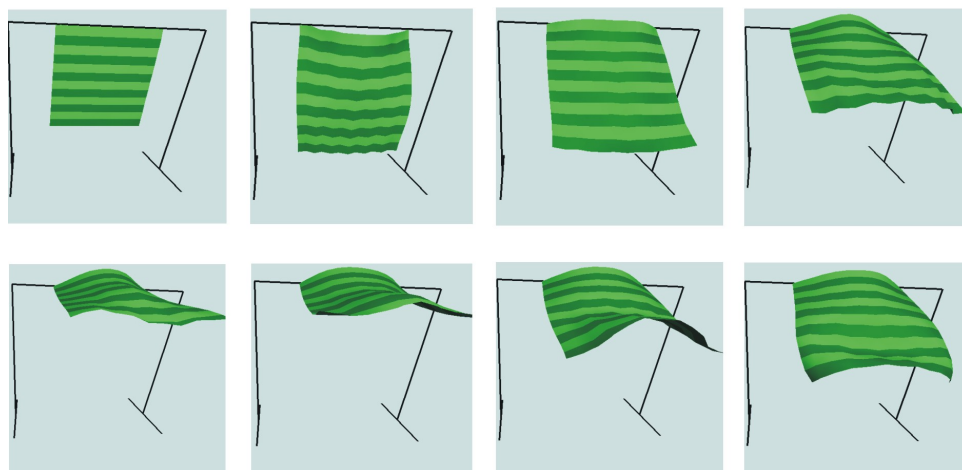


Figura C.2: Vela ao vento com o modelo de malha triangular.

Colisões Para o Modelo de Malha Retangular: Na Figura C.3 exibimos alguns quadros da animação da colisão de um tecido com uma mesa quadrada como mostrada na Figura 5.8(a). Com a malha de 17×17 pontos para o tecido e 33×33 para a mesa, nesse exemplo, obtivemos um tempo de processamento de 36.4 segundos para 3 segundos de animação.

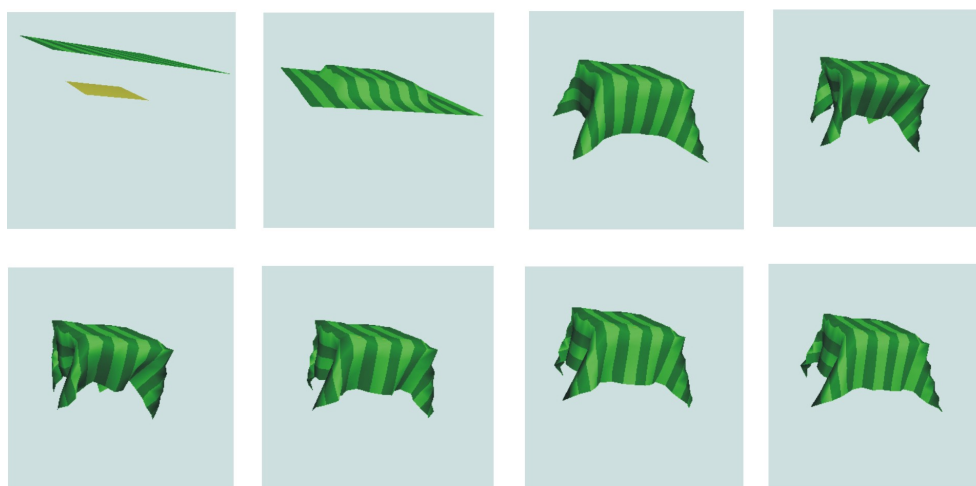


Figura C.3: Colisões para malha retangular: tecido colidindo com mesa quadrada.

Colisões Para o Modelo de Malha Triangular: Na Figura C.4 exibimos alguns quadros da animação da colisão de um tecido com uma mesa quadrada como mostrada na Figura 6.2(a). Com a malha de 17×17 pontos para o tecido e 33×33 para a mesa, nesse exemplo, obtivemos um tempo de processamento de 1 minuto e 7 segundos para 3 segundos de animação.

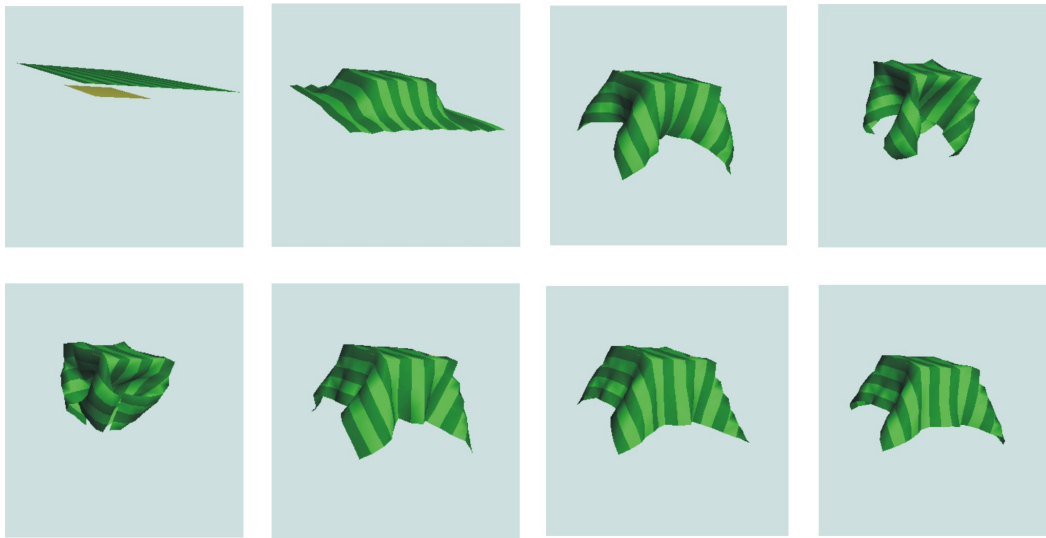


Figura C.4: Colisões para malha triangular: tecido colidindo com mesa quadrada.

Detalhes Implementativos

Neste Apêndice, buscamos expor, de forma sucinta, algumas características de nossa implementação como, por exemplo, a plataforma de desenvolvimento utilizada e o hardware empregado na confecção dos modelos, algoritmos e simulações.

Hardware e Plataforma de Desenvolvimento: O hardware empregado ao longo deste trabalho, tanto na programação quanto na elaboração e execução das simulações, foi um notebook com processador Pentium Celeron M de 1.4GHz com 512 MB de RAM e vídeo Intel 82852 GM. A plataforma de desenvolvimento escolhida foi o Visual C++ 6.0 sobre o sistema operacional Windows XP.

Bibliotecas Gráficas: Toda a etapa de renderização utilizada em nosso estudo ficou a cargo da biblioteca gráfica MESA 3D (<http://www.mesa3d.org>) com controle de janelas baseado na biblioteca GLUT (incluída na MESA 3D).

Implementação: Utilizamos sobrecarga de operadores (uma funcionalidade interessante de C++) para operações com nossos tipos de dados definidos em uma biblioteca matemática que criamos para os modelos e algoritmos de colisões, a *mathadd*. Essa biblioteca contém funções e definições de tipos necessárias às implementações. Desenvolvemos bibliotecas para as estruturas de dados e operações relacionadas a essas estruturas, bibliotecas para os modelos deformáveis e para os algoritmos de colisões. Procuramos, dessa forma, criar códigos reutilizáveis e modulados que simplifiquem a nossa futura tarefa de dar continuidade ao estudo desses modelos.

Referências Bibliográficas

- [1] Y. P. Attencio, C. Esperança, P. R. Cavalcanti and A. A. F. de Oliveira. A Collision Detection and Response Scheme for Simplified Physically Based Animation. In: SIBGRAPI 2005, 2005, Natal.
- [2] D. Baraff and A. Witkin. Large steps in cloth simulation. *Computer Graphics (Proc. SIGGRAPH)*, pages 43-54, 1998.
- [3] D. Baraff and A. Witkin. Physically based modeling. In *SIGGRAPH 2001, Course Notes*, 2001.
- [4] A. Barr. Global and local deformation of solid primitives. In *Proceedings of SIGGRAPH '84*, volume 18, pages 21-29, 1984.
- [5] F. P. Beer, E. Johnston Jr. Mecânica vetorial para engenheiros. 5ª edição - São Paulo: Pearson Education do Brasil, 1994.
- [6] L. Bredley. Numerical solutions of differential equations. Research Report on Curricula and Teaching CT003-98, 1994.
- [7] D.E. Breen, D. House and P. H. Getto. A physically-based particle model of woven cloth. *The Visual Computer*, 8:264-277, 1992.
- [8] D. E. Breen, D. H. House, M. J. Wosny. A particle-based model for simulating the draping behavior of woven cloth. *Textile Research Journal*, Textile Research Institute (TRI/Princeton), 64 (11), pp 663-685, 1994.
- [9] R. Bridson. Computacional aspects of dynamic surfaces. Doctor Thesis, SCCM/Stanford University, 2002.

- [10] R. Bridson, S. Marino and R. Feldkiw. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph. (SIGGRAPH Proc.)*, 21:594-603, 2002.
- [11] R. Bridson, S. Marino and R. Feldkiw. Simulation of clothing with folds and wrinkles. In D. Breen and M. Lin, editors, *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2003.
- [12] M. Carignan, Y. Yang, N. Magnenat-Thalmann and D. Thalmann. Dressing animated synthetic actors with complex deformable clothes. *Computer graphics*, 26(2):99-104, 1992.
- [13] K. -J. Choi and H. -S. Ko. Stable but responsive cloth. *ACM Trans. Gr. (Proc. SIGGRAPH)*, 21:604-611, 2002.
- [14] S. Coquillart. Extend free-form deformation. A sculpturing tool for 3d geometric modeling. *Computer & Graphics*, 24(4):187-196, 1990.
- [15] F. Cordier and N. M. Thalmann. Real time animation of dressed virtual humans. In *EUROGRAPHICS 2002*, 21(3):327-335, 2002.
- [16] C. Cunha. Métodos numéricos. 2a. Edição, Campinas, SP: Editora da UNICAMP, 2003.
- [17] V. Dochev and T. Vassilev. Efficient super-elasticity handling in mass-spring systems. *International Conference on Computer Systems and Technologies - CompSysTech*, 2003.
- [18] V. Dochev and T. Vassilev. Image-space based collision detection in cloth simulation on walking humans. *International Conference on Computer Systems and Technologies - CompSysTech*, 2004.
- [19] P. Faloutsos, M. van de Panne and D. Terzopoulos. Dynamic free-form deformations for animation synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 3(3):201-214, 1997.
- [20] C. Feynman. Modeling appearance of cloth. Master's thesis, Dept. of EECS, Massachusetts Inst. of Technology, Cambridge, Mass., 1986.
- [21] A. Fuhrmann, C. Gross and V. Luckas. Iterative animation of cloth including self collision detection. *Journal of WSCG*, 11(1), February 2003.

- [22] S.F.F. Gibson and B. Mirtich. A survey of deformable modeling in computer graphics. *Technical Report, MERL - A Mitsubishi Electric Research Laboratory*, 1997. Technical Report, <http://www.merl.com> (TR-97-19).
- [23] S. Gottschalk, M. Lin and D. Manocha. OBB tree: A hierarchical structure for rapid interference detection. In *SIGGRAPH 96 Conference Proceedings*, pp 171-180, 1996.
- [24] S. Gottschalk. Collision detection techniques for 3D models. University of North Carolina, 1997.
- [25] D. Halliday, R. Resnick and J. Walker. Fundamentos de física: mecânica - Vol. 1. LTC, Rio de Janeiro, 2003.
- [26] D. House and D.E. Breen. Cloth modeling and animation. *Natick, MA: A. K. Peters*, 2000.
- [27] S. Kawabata. The standardization and analysis of hand evaluation. *The Textile Machinery Society of Japan*, 1975.
- [28] J. Klosowski, M. Held, J. Mitchell, H. Sowizral and K. zikan. Efficient collision detection using bounding volume hierarchies of k-DOPs. *IEEE Transactions on Visualization and Computer Graphics*, 4 (1):21-36, 1998.
- [29] T.L. Kunii, H. Gotoda. Singularity theoretical modeling and animation of garment wrinkle formation process. *The Visual Computer*, Springer Verlag, 6(6), pp 326-336, 1990.
- [30] B. Laffeur, N. Magnenat-Thalmann and D. Thalmann. Cloth animation with self collision detection. *Proc. of Conference on Modeling in Computer Graphics*, Springer, 1991.
- [31] E. L. Lima. Cálculo tensorial. Coleção Notas de Matemática N° 32 - IMPA, 1965.
- [32] K. Loudon. Dominando algoritmos com C. Rio de Janeiro: Editora Ciência Moderna Ltda., 2000.
- [33] V. F. Melo. Modelagem e controle de caimento e dobras em superfícies deformáveis. Tese de Doutorado, FEEC/UNICAMP, 2004.

- [34] L. P. Monteiro e Shin-Ting Wu. Simulação Realista de Tecidos. SBGames 2006, 2006.
- [35] F. I. Nakamura e W. Celes. Detecção hierárquica de colisões em ambientes 3D. *Workshop of Undergraduate Work*, SIBGRAPI, 2005.
- [36] H. Ng and R. Grimsdale. Computer graphics techniques for modeling cloth. *IEEE Computer Graphics and Applications*, 16(5):28-41, 1996.
- [37] X. Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *Proc. of Graphics Interface*, pages 147-154, 1995.
- [38] X. Provot. Collision and self-collision handling in cloth model dedicated to design garments. In *Proc. of Graphics Interface*, pages 177-189, 1997.
- [39] Y. Saad. Iterative methods for sparse linear systems. PWS Publishing. New York, NY, 1996.
- [40] H. Schildt. C, completo e total. São Paulo : Makron, McGraw-Hill, 1990.
- [41] J. C. Simo and D. D. Fox. On a stress resultant geometrically exact shell model. Part ii: The linear theory; computacional aspects. *Computer Methods in Applied Mechanics and Engeneering*, 73:53-92, 1989.
- [42] D. Terzopoulos, J. C. Platt, A. H. Barr and K. Fleischer. Elastically deformable models. *Computer Graphics*, 21(4):205-214, 1987.
- [43] G. Van Den Bergen. Efficient collision detection of complex deformable models using AABB tres. *Journal of Graphic Tools*, 2,4 (1997), 1-14, 1997.
- [44] A. Van Gelder. Aproximate simulation of elastic membranes by triangulated spring meshes. *Journal of Graphics Tools*, 3(2):21-42, 1998.
- [45] T. Vassilev and B. Spanlang. Efficient cloth model for dressing animated virtual people. In *Proc. Learning to Behave Workshop*, pages 89-100. Enschede the Netherlands, 2000.
- [46] P. Volino and N. Magnenat-Thalmann. Virtual clothing. Theory and practice. *Berlin: Springer*, 2000.
- [47] P. Volino and N. Magnenat-Thalmann. Efficient self collision detection on smoothly discretized surface animations using geometrical shape regularity. In

Computer Graphics Forum (EuroGraphics Proc.), Volume 13, pages 155-166, 1994.

- [48] P. Volino and N. Magnenat-Thalmann. Developing simulation techniques for an interactive clothing system. In *Pro. of the 1997 International Conf. on Virtual Systems and Multimedia*, pages 109-118. IEEE, 1997.