

Simulação Realista de Tecidos

Leandro P. Monteiro Shin-Ting Wu

Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e Computação, Brasil

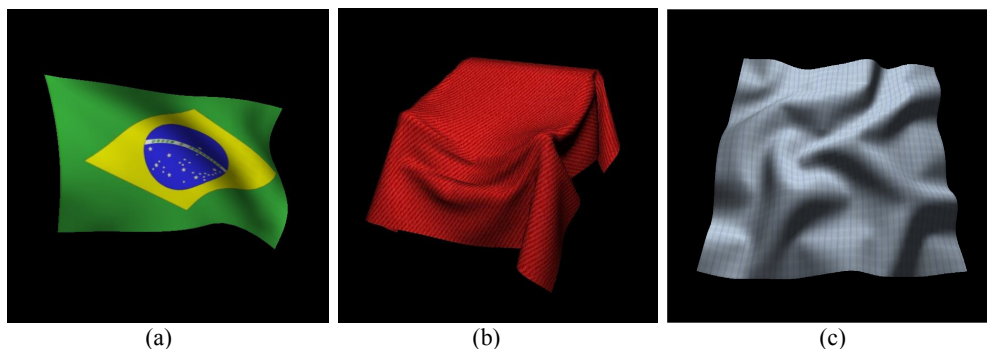


Figura 1: (a) bandeira hasteada, (b) poltrona coberta, (c) pano amassado.

Resumo

A crescente busca pelo realismo nos jogos tem levado à integração da simulação de tecidos para fazer os movimentos de vestuários, sobre um personagem animado, mais próximos da nossa percepção usual. Este artigo objetiva apresentar a eficácia de um paradigma geometricamente exato, fundamentado na teoria de superfície de Cosserat Elástica, para simulação de tecidos, descrevendo uma implementação com integração temporal explícita, de baixo custo computacional. Dessa forma, nossa implementação do modelo deformável resulta em comportamentos físicos convincentes, mesmo usando integrações temporais simples, como as integrações Euler ou Verlet.

Palavras-chaves: modelagem e animação de tecidos, modelagem embasada fisicamente, simulação de tecidos

Contato dos autores:

{lpm,ting}@dca.fee.unicamp.br

1. Introdução

Os jogos e os principais programas para a construção de cenas 3D, como 3DMax e Maya, têm se beneficiado de modelos embasados na Física para gerar cenários impressionantes. Boa parte do notável crescimento no realismo dos jogos deve-se à grande utilização de modelos físicos dinâmicos. Ao invés da geometria estática dos jogos antigos, é possível notar, nos atuais, que os objetos se comportam realisticamente diante de diversos tipos de força, como a gravidade e a resposta aos impactos com outros objetos. Muitos destes avanços estão relacionados diretamente com as novas tecnologias de placas gráficas, que surgem cada vez mais velozes realizando cálculos complexos diretamente na GPU [Kolb e Cuntz 2005]. Uma das fronteiras das simulações físicas é a simulação de tecidos, da qual se espera a modelagem de tecidos em tempo real com um alto grau de realismo. Tais

simulações são necessárias para modelar ao longo de um jogo, por exemplo, as roupas de personagens computadorizados em movimento ou uma bandeira hasteada ondulando.

Há pouco tempo atrás a simulação de tecidos nos jogos estava restrita às cenas pré-processadas. Um dos primeiros jogos que incorporou algumas técnicas simples de simulação em tempo real de tecidos foi “*Hitman: Code Name 47*” (IO Interactive, 2000), o qual utilizava o modelo proposto por Jakobsen [2001] para simular deformações de casacos e capas, porém de forma pouco convincente. Atualmente, a tendência são os jogos massivos multi-jogador (MMOG¹), capazes de conectar centenas de jogadores que interagem entre si em um mesmo mapa. Dessa forma, a velocidade no cálculo e recebimento de cada quadro necessita ser altíssima, o que muitas vezes desencoraja o projetista do jogo a incluir uma simulação de tecidos mais realista para não deteriorar a jogabilidade e a escalabilidade.

A dificuldade em modelar um tecido reside no seu peculiar comportamento, o qual é ser muito permissivo à variação de curvatura e restritivo à variação de área. Com a exceção dos tecidos feitos de elastano, que possuem uma variação de área maior, os demais apresentam, sob deformações razoáveis, uma mudança de tamanho quase imperceptível. Um tecido é uma estrutura complexa composta de fios, entrelaçados em um determinado padrão. Os fios são feitos de fibras, naturais como a lã e o algodão ou sintéticas como o elastano e o couro, que determinam o material do tecido. A sua massa e o seu comportamento estão diretamente relacionados ao material, à composição, ao padrão de entrelaçamento e ao tratamento de seus fios.

Devido a essa diversidade de parâmetros a modelagem de um tecido pode ser feita de muitas formas. Existem três metas relacionadas à simulação

¹ *Massive Multiplayer Online Games*

de tecidos que são colocadas em primeiro plano dependendo da finalidade da aplicação: o realismo, em animações, como filmes, desenhos e propagandas; a eficiência, relevante nos jogos em geral, pois de nada adianta um jogo realista se sua interatividade ficar prejudicada por algum método custoso que, na prática, serve apenas para lhe dar um maior adereço; e a precisão física, importante para análise e simulação do comportamento de tecidos nas aplicações da Engenharia Têxtil.

Os modelos embasados na Física são reconhecidos por serem os mais promissores para produzir uma aparência natural do movimento de tecidos. Um tecido é representado como uma superfície deformável cuja dinâmica é governada pela equação diferencial parcial de equilíbrio em cada um de seus pontos \mathbf{r} [Terzopoulos et al. 1987]

$$\mu \frac{\partial^2 \mathbf{r}}{\partial t^2} + \gamma \frac{\partial \mathbf{r}}{\partial t} + \frac{\delta \mathcal{E}(\mathbf{r})}{\delta \mathbf{r}} = f(\mathbf{r}, t), \quad (1)$$

na qual μ é a densidade de massa relacionada a energia dinâmica, γ é a densidade de amortecimento que reduz a energia cinética, $f(\mathbf{r}, t)$ representa a contribuição total das forças externas em \mathbf{r} em um dado instante t , e o termo $\delta \mathcal{E}(\mathbf{r})/\delta \mathbf{r}$ corresponde à energia interna acumulada na superfície.

É no último termo, referente à energia interna, que se diferenciam duas abordagens dentro dos modelos físicos: a mecânica de partículas e a mecânica de contínuo. A mecânica de partículas considera cada elemento isoladamente, em um conjunto discreto, aproximando a energia interna de cada ponto pelas tensões entre as partículas vizinhas [Breen et al. 1994]. Já a mecânica de contínuo considera um conjunto contínuo e estabelece, para cada ponto da superfície, uma energia interna em função dos parâmetros da geometria diferencial [Terzopoulos et al. 1987].

Ambas abordagens buscam solucionar um mesmo problema, que é o de encontrar de forma precisa e estável, a partir de um instante t , as novas posições dos pontos que representam o tecido, depois de um instante $t + \Delta t$, no qual Δt é o chamado passo de tempo. Esta integração temporal corresponde à dinâmica dos tecidos e consiste na resolução da equação (1), com o vetor \mathbf{r} sendo a incógnita. Há três paradigmas distintos para fazer isso, os quais se diferem entre si pela precisão, pela estabilidade e pela eficiência: explícito, semi-implícito e implícito.

A integração implícita, sob o ponto de vista teórico, é a mais correta porque integra os dois lados da equação (1) em relação ao tempo e o sistema deve ser resolvido pela solução das duas integrais, levando-se em consideração os novos dados da mesma iteração, como as forças resultantes e os elementos referentes às partes cinemática e dinâmica. Já a integração explícita

isola o vetor \mathbf{r} na equação (1) e encontra as novas posições com base em todos os dados da iteração anterior. Por fim, a integração semi-implícita seria uma mistura das duas, pois utiliza parte dos dados da iteração anterior e parte da atual, formulando estes últimos para serem encontrados na mesma iteração dos novos pontos. A integração semi-implícita, por mesclar a precisão e estabilidade do esquema implícito com a eficiência do explícito, junto com a mecânica de partículas, mais especificamente o modelo massa-mola, formam a tendência atual da simulação de tecidos [Baraff et al. 2003; Bridson et al. 2003].

Nossa proposta visa uma nova abordagem que una a mecânica de contínuo com a integração explícita. Com isso, acreditamos que a precisão perseguida pelas integrações implícitas e semi-implícitas será encontrada na formulação geometricamente exata da superfície de Cosserat Elástica, enquanto a integração explícita proporcionará a eficiência necessária para os jogos. Assim, na seção 2 são citados os trabalhos relacionados com o intuito de alocar nosso trabalho em uma lacuna existente na literatura. A seção 3 apresenta o modelo de superfície deformável [Wu e Melo 2004] utilizado como base deste trabalho, e sua discretização espacial e temporal. Os problemas e as limitações deste modelo, bem como as respectivas soluções propostas, como a integração explícita, serão o tema da seção 4. Na seção 5 são mostrados e analisados os resultados obtidos para, por fim, tecer conclusões e futuros desdobramentos na seção 6.

2. Trabalhos Relacionados

Um dos primeiros trabalhos da mecânica de contínuo pertence a Feynman [1986], no qual ele encontrava a força interna da superfície minimizando as energias de tensão e curvatura. Neste trabalho precursor, ele utilizava a integração temporal implícita com um método de relaxação. Pouco depois surgiu o trabalho de Terzopoulos et al. [1987], que também utilizava a mecânica de contínuo, porém baseava seu modelo na teoria da elasticidade, fazendo com que a força interna fosse aproximada pelas variações das medidas geométricas ao longo do tempo relacionadas aos coeficientes elásticos da superfície. Neste trabalho ele utilizou uma integração temporal semi-implícita com o método dos gradientes conjugados, conseguindo uma significativa contribuição para a área na época, porém o modelo apresentava muitos problemas devido a uma consideração equivocada na energia interna e uma estimativa ruim para o vetor normal. Foi então que surgiu o trabalho de Wu e Melo [2004; Melo 2004] que, fundamentado em um caso particular da teoria de superfície de Cosserat Elástica, conseguiu uma melhor estimativa para o vetor normal e para a energia interna. Este trabalho adotou o mesmo esquema de integração temporal semi-implícita proposto por Terzopoulos et al. [1987].

No outro ramo, da mecânica de partículas, o trabalho precursor é de Haumann [1987], que desenvolveu o primeiro modelo massa-mola para modelar a dinâmica de uma superfície deformável, utilizando a integração temporal explícita. Neste modelo cada nó da superfície é conectado ao seu vizinho por uma mola e os coeficientes destas determinam a energia interna. Depois deste trabalho Breen et al. [1994] desenvolveram um trabalho original, sem usar o modelo massa-mola, que contabilizava a força interna da superfície como um somatório de três tipos de energia baseadas nas distâncias e nos ângulos entre as partículas adjacentes. Eles também utilizaram a integração explícita. Por fim, Provot [1995] construiu um modelo massa-mola estendido em relação ao de Haumann usando três tipos de mola entre partículas vizinhas em uma vizinhança de 24 nós adjacentes, resolvendo alguns problemas encontrados na época, como o da superelasticidade. Este modelo também utilizava a integração explícita e foi amplamente difundido por sua simplicidade e eficiência, apesar de efeitos ainda aquém de serem convincentes.

A partir desse ponto, os autores começaram a integrar as vantagens da mecânica de partículas, mais especificamente do modelo massa-mola, com uma integração temporal precisa, como a implícita e semi-implícita. Um trabalho amplamente conhecido é o de Baraff e Witkin [1998], no qual utilizaram a modelagem de Provot junto com uma integração temporal implícita, que resultou em simulações relativamente rápidas para a época e ainda com um passo de tempo maior do que os utilizados até então. Na mesma linha, mas utilizando uma integração temporal semi-implícita, pode-se citar Bridson et al. [Bridson et al. 2002; Bridson et al. 2003], os quais isolaram a dinâmica dos tecidos e elaboraram, sobre ela, um algoritmo de detecção, tratamento e resposta a colisões muito eficiente. Nesta integração era usado um esquema explícito para encontrar forças internas e implícito para as dissipadoras. O último trabalho deste campo e bastante atual, pertence mais uma vez a Baraff et al. [2003], no qual utilizando seu modelo anterior desenvolve algumas técnicas para desembaraçar vestuários, colhendo resultados bem convincentes.

Os trabalhos que tentaram aplicar essas técnicas em tempo real surgiram impulsionados pelo trabalho de Baraff e Witkin [1998], pois com um passo de tempo maior podiam-se ter menos iterações, as quais eram custosas, entre um instante e outro da simulação em tempo real. O trabalho de Desbrun et al. [1999] visava ser uma solução em tempo real, pois se tratava de Realidade Virtual, porém não tinha o apelo da eficiência como nos jogos. O trabalho de Oshita e Makinouchi [2001] utilizava um modelo massa-mola de poucos vértices com um algoritmo para refinar e triangular as faces da superfície e, junto com a integração semi-implícita, obteve resultados visuais não muito convincentes de uma saia sobre um personagem animado. Cordier e Thalmann [2002]

apresentaram otimizações para tratar colisões de vestuários também sobre um modelo animado. Podem-se ainda citar as teses de Boxerman [2003] e Bargmann [2003], ambos trabalhos em tempo real de modelos massa-mola, inspirados no passo de tempo largo de Baraff e Witkin [1998], com diversas contribuições sobre colisões com outros objetos e autocolisões.

Apesar da eficiência alcançada com os modelos de partículas, simulações de alguns tipos de comportamento dos tecidos estão ainda aquém do realismo. Um destes efeitos é o movimento perpendicular ao plano sob forças puramente tangenciais que, de nosso conhecimento, somente foram avaliados de forma natural nos trabalhos de Wu e Melo [2004; Melo 2004]. Eles consideraram em seu modelo deformável um coeficiente de energia interna que relaciona as deformações métricas com as de curvatura, reproduzindo dobras e vincos sem mudança de área e sem adição de forças artificiais para conseguir o mesmo efeito [Volino e Thalmann 1999]. Apresentamos neste artigo uma implementação do modelo de Wu e Melo [2004] no esquema de integração explícita, pois acreditamos que nela a precisão física da modelagem seja mantida, enquanto a substituição da integração temporal semi-implícita para a explícita forneça a eficiência requerida para os jogos.

3. Modelo Deformável

Esta seção apresenta uma breve descrição do modelo proposto por Wu e Melo [2004], mais detalhado em [Melo 2004], e sua implementação no esquema de integração temporal semi-implícita, mostrada pelos mesmos autores. Maiores informações constam na página do projeto DeSMo (*Deformable Surface Modeling*), que pode ser acessada pelo endereço eletrônico: www.dca.fee.unicamp.br/projects/desmo.

O modelo apresentado em [Wu e Melo 2004; Melo 2004] é fundamentado na teoria de Superfície Cosserat, a qual foi originalmente proposta pelos irmãos Cosserat em 1909, redescoberta na década de 50 para modelagem de corpos orientados e, mais tarde, usada para modelagem de cascas por Green et al. [1965]. Em 1989 Simo e Fox [1989] apresentaram uma solução numérica computacionalmente factível para tal modelo e, em 1996, Eischen et al. [1996] aplicaram este esquema numérico para modelar a dinâmica dos tecidos.

A superfície de Cosserat é uma superfície regular parametrizada, ao longo de duas curvas coordenadas, que possui associado a cada ponto um vetor que aponta para fora do plano tangente, chamado vetor diretor. Supõe-se um domínio de coordenadas bidimensionais x^1 e x^2 , e uma função posição $\mathbf{r}(x^1, x^2)$, que mapeie os pontos deste domínio para pontos tridimensionais $(\mathbf{x}(x^1, x^2), \mathbf{y}(x^1, x^2), \mathbf{z}(x^1, x^2))$. Desta forma, as derivadas parciais $\partial\mathbf{r}/\partial x^1$ e $\partial\mathbf{r}/\partial x^2$ da função \mathbf{r} representam dois vetores linearmente independentes, os quais junto com

o vetor diretor \mathbf{d} , formam um sistema de referência local em cada ponto da superfície S (figura 2).

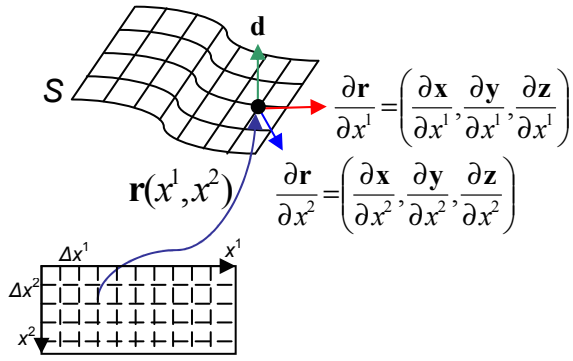


Figura 2: Parametrização da superfície de Cosserat.

No seu trabalho, Wu e Melo [2004] propuseram utilizar o vetor normal unitário do ponto à superfície como o seu vetor diretor \mathbf{d}

$$\mathbf{d} = \mathbf{n} = \frac{\frac{\partial \mathbf{r}}{\partial x^1} \times \frac{\partial \mathbf{r}}{\partial x^2}}{\left| \frac{\partial \mathbf{r}}{\partial x^1} \times \frac{\partial \mathbf{r}}{\partial x^2} \right|}. \quad (2)$$

Além desses elementos que definem o referencial local a cada ponto, existem duas medidas geométricas muito importantes para o modelo, denominadas de tensores métrico e de curvatura. O tensor métrico $G_{\alpha\beta}$ de cada ponto é o produto escalar entre seus vetores de base e está relacionado com a área na vizinhança do ponto

$$G_{\alpha\beta} = \frac{\partial \mathbf{r}}{\partial x^\alpha} \cdot \frac{\partial \mathbf{r}}{\partial x^\beta}, \quad \alpha, \beta = 1, 2. \quad (3)$$

O tensor de curvatura $B_{\alpha\beta}$ de cada ponto é o produto entre a derivada segunda multiplicada pelo vetor normal e indica o quanto o ponto está curvado na direção do vetor normal

$$B_{\alpha\beta} = \mathbf{n} \cdot \frac{\partial^2 \mathbf{r}}{\partial x^\alpha \partial x^\beta}, \quad \alpha, \beta = 1, 2. \quad (4)$$

Os tensores variam, em cada iteração da dinâmica, de acordo com a deformação que a superfície sofre. Estas variações em relação aos valores $G_{\alpha\beta}^0$ e $B_{\alpha\beta}^0$, da superfície inicial S_0 , definem duas medidas físicas:

1) variação da métrica

$$\varepsilon_{\alpha\beta} = G_{\alpha\beta} - G_{\alpha\beta}^0, \quad \alpha, \beta = 1, 2, \quad (5)$$

2) variação da curvatura

$$\kappa_{\alpha\beta} = B_{\alpha\beta} - B_{\alpha\beta}^0, \quad \alpha, \beta = 1, 2. \quad (6)$$

A energia interna E armazenada em uma superfície de Cosserat, correspondente ao terceiro termo do lado esquerdo na equação (1), é expressa em termos dessas variações

$$E = \sum_{\alpha, \beta=1}^2 [\Phi^{\alpha\beta} \varepsilon_{\alpha\beta} \varepsilon_{\alpha\beta} + \Psi^{\alpha\beta} \kappa_{\alpha\beta} \kappa_{\alpha\beta} + \Theta^{\alpha\beta} \varepsilon_{\alpha\beta} \kappa_{\alpha\beta}], \quad (7)$$

na qual os coeficientes Φ e Ψ estão diretamente associados ao material de que é feito o tecido ou, mais genericamente, à superfície deformável. Podem-se definir duas características de um tecido a partir destes coeficientes, que são a resistência à mudança de métrica (Φ) e a resistência à mudança de curvatura (Ψ). O terceiro coeficiente Θ é chamado de termo de acoplamento e estabelece a variação da curvatura na vizinhança de um ponto sem que a sua área seja alterada. Este termo faz com que simulações de *buckling*² e *folding*³ sejam muito mais realistas e intuitivas, não necessitando de forças dissipadoras na direção perpendicular às aplicadas no movimento da superfície [Volino e Thalmann 1999].

Com base nesse modelo, Wu e Melo apresentaram um algoritmo que tem como entrada três grupos de informações: parâmetros de superfície, de ambiente e de simulação. A figura 3 mostra as cinco etapas da simulação de tecidos que relacionam os elementos do modelo. O início do fluxo parte dos dados de entrada para gerar a superfície inicial, a partir da qual as medidas geométricas são calculadas. Em seguida, com a variação destas últimas são definidas as medidas físicas que, por sua vez, fazem parte da energia interna armazenada. Por fim, a etapa principal consiste no uso desses dados na equação de equilíbrio (1) para obter as novas posições dos pontos da superfície.

Para solucionar numericamente a equação (1), Wu e Melo [2004] utilizaram o mesmo paradigma de discretização espacial e temporal que Terzopoulos et al. propuseram [1987].

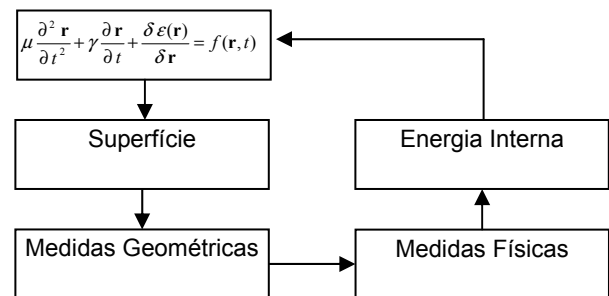


Figura 3: Fluxograma da dinâmica da superfície.

² Simulações de um pedaço de tecido amassado, com a formação de dobras, vincos ou rugas, que normalmente são geradas usando-se adição de forças na direção perpendicular à superfície.

³ Simulações de um pedaço de tecido dobrado, normalmente consistindo de poucas dobras uniformes.

A primeira preocupação da discretização espacial é tornar o domínio contínuo do modelo apresentado em um conjunto discreto de nós bidimensionais. Dessa forma, a superfície é discretizada por uma malha retangular de $m \times n$ pontos, dispostos em duas direções, x^1 e x^2 , cujos espaçamentos entre nós consecutivos são Δx^1 e Δx^2 , respectivamente (figura 4).

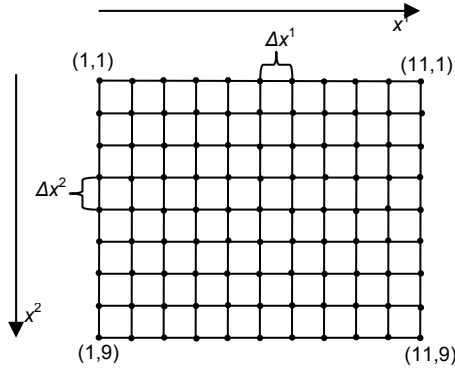


Figura 4: Grade de 11x9 representando uma superfície.

A segunda preocupação seria aproximar as derivadas de primeira e de segunda ordem. Para isso, foi utilizado o método das diferenças finitas, conforme descrito em [Smith 1985]. Neste método, cada nó $[k, l]$, sendo $k = 1..m$ e $l = 1..n$, refere-se a um ponto discreto no domínio de acordo com a convenção da figura 4. Considerando uma função malha ou vetorial, são definidos operadores avançados, atrasados e centrais, em uma determinada ordem, de acordo com direção em que os pontos vizinhos são utilizados para aproximar as derivadas desejadas.

Wu e Melo [2004] consideraram os operadores de primeira ordem avançados para definir a derivada primeira das funções utilizadas no modelo. Os vetores de base, utilizados para o cálculo de outros elementos como o tensor métrico mostrado em (3), são expressos por

$$\mathbf{r}_1[k, l] = D_1^+ \mathbf{r}[k, l] = \frac{\mathbf{r}[k+1, l] - \mathbf{r}[k, l]}{\Delta x^1}, \quad (8)$$

$$\mathbf{r}_2[k, l] = D_2^+ \mathbf{r}[k, l] = \frac{\mathbf{r}[k, l+1] - \mathbf{r}[k, l]}{\Delta x^2}. \quad (9)$$

Para a segunda derivada também são utilizados os operadores avançados, para os índices cruzados (equação 10, sendo $\mathbf{r}_{1,2} = \mathbf{r}_{2,1}$), e centrais, para índices iguais (equações 11 e 12), conforme definido nas expressões

$$\mathbf{r}_{1,2}[k, l] = \frac{\mathbf{r}[k+1, l+1] - \mathbf{r}[k+1, l] - \mathbf{r}[k, l+1] + \mathbf{r}[k, l]}{\Delta x^1 * \Delta x^2}, \quad (10)$$

$$\mathbf{r}_{1,1}[k, l] = \frac{\mathbf{r}[k+1, l] - 2 * \mathbf{r}[k, l] + \mathbf{r}[k-1, l]}{\Delta x^1 * \Delta x^1}, \quad (11)$$

$$\mathbf{r}_{2,2}[k, l] = \frac{\mathbf{r}[k, l+1] - 2 * \mathbf{r}[k, l] + \mathbf{r}[k, l-1]}{\Delta x^2 * \Delta x^2}. \quad (12)$$

A discretização temporal, por sua vez, estabelece que o tempo será integrado de forma semi-implícita para construir a dinâmica da superfície. Assim, a equação (1), aplicável a cada ponto da superfície, é transformada em um sistema matricial contendo os novos pontos a serem encontrados em cada iteração, bem como alguns elementos equacionados em função destes pontos, como parte da energia interna E decomposta por

$$E = \tilde{E} + \hat{E}, \quad (13)$$

sendo \tilde{E} a parcela envolvendo termos de ordem maior, a qual é aproximada com base nos elementos da iteração anterior, e \hat{E} a parcela referente à iteração atual, possuindo somente termos linearizáveis e expressa por

$$\hat{E} = K \mathbf{R}, \quad (14)$$

em que K é a matriz de rigidez, de ordem $mn \times mn$, que estabelece a tensão entre cada par de nós, e \mathbf{R} é o vetor coluna contendo os pontos a serem encontrados.

De posse dessa última expressão e usando a energia interna E decomposta em (13), pode-se reescrever a equação (1) transformando-a no sistema matricial

$$M \frac{\partial^2 \mathbf{R}}{\partial t^2} + C \frac{\partial \mathbf{R}}{\partial t} + K \mathbf{R} = F - \tilde{E}, \quad (15)$$

sendo M a matriz diagonal formada pelas densidades de massa, C a matriz diagonal formada pelos coeficientes de amortecimento, K a matriz de rigidez referente à energia interna do instante $t + \Delta t$, F a matriz-coluna contendo a força externa resultante no instante t , \tilde{E} a matriz-coluna da energia interna referente ao instante t e \mathbf{R} é a matriz-coluna incógnita do sistema que contém as posições dos pontos da superfície da próxima iteração $t + \Delta t$.

Assim, o sistema expresso pela equação (15) deve ser solucionado em cada nova iteração $t + \Delta t$ da simulação. Observe que a ordem da matriz quadrada K é mn , ou seja, sempre da ordem do número de vértices da malha elevado ao quadrado. Dessa forma, a solução do sistema em cada iteração possui um custo computacional de ordem exponencial que, neste caso, é da ordem $O((mn)^2)$. Para os jogos, tal custo inviabiliza o uso de modelos complexos.

4. Integração Explícita

Nesta seção apresentamos um esquema alternativo de integração temporal para solucionar a equação (1). O esquema de integração explícita que adotamos é detalhado na seção 4.1. Na seção 4.2 discutiremos os problemas e soluções relacionadas com a definição de condições de fronteira que encontramos ao analisarmos a discretização espacial apresentada na seção 3.

4.1 Integração Euler

A integração temporal explícita é também conhecida como integração de Euler para frente e, como o nome mesmo diz, explicita a solução em função dos termos conhecidos. Desta forma, partindo da equação (1) e isolando-se a parcela referente à aceleração, obtém-se

$$\mu \frac{\partial^2 \mathbf{r}}{\partial t^2} = F(\mathbf{r}, t) - \gamma \frac{\partial \mathbf{r}}{\partial t}, \quad (16)$$

sendo $F(\mathbf{r}, t)$ a força resultante entre as externas e internas. Assim, o primeiro passo de cada iteração consiste em encontrar a aceleração resultante dos pontos $[k, l]$ da superfície após Δt , com base nos termos da iteração anterior

$$a_{k,l}(t + \Delta t) = \frac{1}{\mu} F_{k,l}(t) - \gamma v_{k,l}(t). \quad (17)$$

Após a nova aceleração ser encontrada, a velocidade dos pontos é atualizada

$$v_{k,l}(t + \Delta t) = v_{k,l}(t) + \Delta t a_{k,l}(t + \Delta t), \quad (18)$$

para, por fim, atualizar suas posições

$$\mathbf{r}_{k,l}(t + \Delta t) = \mathbf{r}_{k,l}(t) + \Delta t v_{k,l}(t + \Delta t). \quad (19)$$

Esse esquema é bastante simples e rápido, pois nele não é preciso solucionar um sistema de equações lineares nem reescrever outros termos em função dos novos pontos. O processo é usado para encontrar as novas posições de cada elemento separadamente, no instante $t + \Delta t$, com base nos dados da iteração t . Isto pode ser feito desde que se considere que a superfície varie suavemente entre duas iterações subsequentes para que o erro de aproximação seja pequeno, ou seja, ao se usar os dados do instante anterior ao invés do atual deseja-se que a variação destes valores seja a menor possível. Para garantir essa suavidade em todas as simulações deste artigo o passo de tempo utilizado foi de 0,001 segundos. Dessa forma, se no modelo de Wu e Melo [2004] com a integração semi-implícita, o passo de tempo utilizado era de 0,01 segundos, ou seja, 100 iterações para se simular 1 segundo de animação, neste novo esquema são necessárias 1000 iterações. Contudo, nesse esquema atual temos um custo linear de ordem $O(mn)$ contra $O((mn)^2)$, afinal, aqui já não é mais preciso utilizar a matriz de rigidez. Na seção 5 será mostrado o tempo gasto por iteração de cada simulação para que este ganho de eficiência fique nítido.

Um dos grandes problemas relacionados atualmente com o uso da dinâmica de tecidos em tempo real é o chamado problema de escalabilidade. Tal problema refere-se ao descompasso entre o passo

de tempo usado na dinâmica dos tecidos para o passo de tempo da simulação em tempo real. Naturalmente, em apenas um ciclo de tempo da simulação em tempo real, como nos jogos, é preciso realizar toda a dinâmica de tecido correspondente a este intervalo, exigindo que o mínimo de processamento seja usado para que os demais cálculos não sejam afetados e não se gere um atraso entre os quadros da simulação. Devido a esse problema e sabendo que, teoricamente, em uma simulação em tempo real aproximadamente 30 quadros por segundo são suficientes para manter a idéia de continuidade entre os movimentos, as iterações intermediárias da dinâmica dos tecidos entre os quadros da simulação precisam ser geradas da forma mais rápida possível. Neste sentido, os métodos de integração explícita são os mais indicados de serem usados, desde que o passo de tempo imposto por uma aproximação estável de $F(\mathbf{r}, t)$ seja respeitado.

Integrar explicitamente um modelo massa-mola tem a vantagem da complexidade de cômputo de cada quadro ser linear, porém requer um passo de tempo muito pequeno para assegurar a estabilidade no cálculo de $\delta \mathcal{E}(\mathbf{r})/\delta \mathbf{r}$. Embora Baraff e Witkin [1998] tenham mostrado que o uso da integração implícita baseada nos métodos de gradiente conjugado permite aumentar este passo de tempo, o valor apresentado por eles no seu artigo é somente 10 vezes maior do que o passo utilizado nas nossas simulações. Acreditamos que isso decorre do fato do modelo geometricamente exato proposto por Wu e Melo [2004] aproximar melhor a dinâmica de tecidos. Conjugando a complexidade linear do método de integração explícita com o uso de um passo de tempo uma ordem menor ao que foi utilizado por Baraff e Witkin [1998], conseguimos atingir um desempenho superior, em termos de custo computacional, aos trabalhos do nosso conhecimento.

Como uma alternativa de comparação do método numérico adotado neste trabalho, mais próxima do esquema explícito do que o antigo semi-implícito, implementamos também a integração Verlet usada em Jakobsen [2001]. Nesta integração os dois últimos passos expressos em (18) e (19) da integração Euler são substituídos por

$$\mathbf{r}_{k,l}(t + \Delta t) = 2 * \mathbf{r}_{k,l}(t) - \mathbf{r}_{k,l}(t - \Delta t) + a \Delta t^2. \quad (20)$$

Dessa forma, usando dois vetores que armazenam as posições dos pontos da superfície, corrente e anterior, o esquema de evolução do tempo é feito sem o cálculo direto da velocidade.

A diferença entre as integrações de Euler e Verlet acontece na precisão, pois em Verlet o erro das soluções diminui por aproximar a velocidade do próximo instante $t + \Delta t$ pelo intervalo $[t - \Delta t, t + \Delta t]$ ao invés de $[t - \Delta t, t]$, como na integração Euler. Entretanto, os resultados visuais das duas implementações foram similares ao passo de tempo imposto por $\delta \mathcal{E}(\mathbf{r})/\delta \mathbf{r}$ para a sua convergência.

Acreditamos que esta diferença de precisão só afete visualmente os resultados à medida que se use um passo de tempo maior que o empregado neste modelo, pois assim como no trabalho de Baraff e Witkin [1998], Jakobsen [2001] também usa um modelo de partículas, ao invés de um modelo físico preciso, para aproximar uma superfície deformável. Contudo, é interessante salientar que o modelo deformável utilizado apresentou o mesmo desempenho e os mesmos resultados visuais em ambas as integrações, mostrando que o realismo e a precisão física são atributos alcançados pelo modelo de tecido usado, e não pelo método numérico escolhido.

4.2 Condições de Fronteira

No primeiro instante, utilizamos o mesmo esquema de diferenças finitas para computar os componentes da energia interna E definida por (7). No entanto, constatamos que os operadores escolhidos para discretizar os elementos do modelo levam a um problema nas bordas, pois nem sempre ter-se-á os nós necessários para os operadores usados. Para estes casos, em que faltam os nós vizinhos, as derivadas de primeira e segunda ordem são zeradas, de acordo com as condições de fronteira escolhidas na implementação de Wu e Melo [2004].

Mais especificamente, usando a notação da figura 5, as bordas B1 e B2 possuem todos os pontos da primeira derivada bem definidos, enquanto que as bordas B3 e B4 possuem derivadas nulas nas direções x^1 e x^2 , respectivamente, pois não possuem os nós $[k+1, l]$ para o operador definido por (8) e $[k, l+1]$ para o (9). Para a derivada segunda todas as bordas utilizam-se, para os operadores centrais definidos por (11) e (12), de um ponto à frente e um atrás e, portanto, todos os nós das quatro bordas possuem derivadas nulas. No operador cruzado avançado (10) novamente o problema acontece apenas nas bordas B3 e B4, pois nelas os vizinhos $[k+1, l]$, $[k, l+1]$ e $[k+1, l+1]$ nunca existem em um mesmo nó.

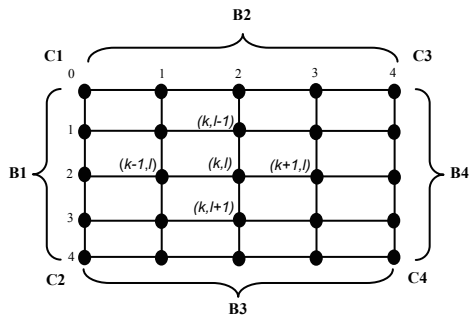


Figura 5: Notação de bordas e cantos.

Na figura 5 o ponto (2,2) está exemplificado como nó $[k, l]$ para que fique clara a relação dos pontos com seus vizinhos por meio dos índices da malha. Naturalmente, o canto C2 sofre o mesmo problema da borda B3, assim como C3 o de B4 e, ainda, C4 o de ambas.

Dessa forma, tais condições podem resultar, com frequência, em valores desproporcionais da força interna nas bordas, o que afeta toda dinâmica da superfície. A figura 6(a) exemplifica o aspecto visual de um caso em que o valor da força interna de um ponto da borda tende ao infinito. Além do problema de instabilidade numérica na simulação, há também casos em que o mau condicionamento na fronteira provoca imprecisão em algumas bordas, conforme a figura 6(b), em que é possível notar as duas bordas extremas com aspectos visuais ruins. A solução que visa atenuar tais problemas é apresentada a seguir.

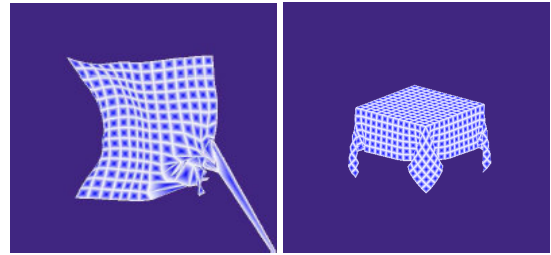


Figura 6: Resultados inesperados nas bordas.

O novo tratamento usado nas bordas é bastante intuitivo, pois ao invés de zerar as derivadas onde os pontos necessários não existirem são pegos pontos na direção em que eles existem, ou seja, nenhum nó da malha possuirá a derivada primeira e segunda nula.

Portanto, para a primeira derivada ter-se-ia a convenção utilizada na figura 7: para os pontos até os índices $m-1$ e $n-1$, no caso índice 3 em ambas as direções, são usados os antigos operadores, como mostra a parte inferior direita da figura; já para os pontos das bordas B3 e B4 são usados os operadores atrasados de primeira ordem, conforme a parte superior esquerda da mesma figura.

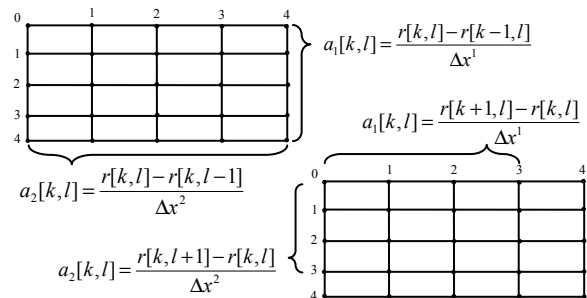


Figura 7: Nova convenção para a derivada primeira.

A mesma idéia é estendida para a derivada segunda, sendo que para os pontos fora das bordas da superfície o cálculo continuaria o mesmo, como mostra a figura 8, e para os da borda seriam usados os pontos existentes, conforme a convenção da figura 9.

Assim, no canto C1, que antes possuía o operador central nulo em ambas as direções, agora terá valores com os operadores avançados, usando os nós $[k+1, l]$ e $[k+2, l]$ na direção x^1 , e $[k, l+1]$ e $[k, l+2]$ na direção x^2 . A mesma consideração é usada no canto C4, ou seja,

ao invés dos operadores centrais são usados os das direções existentes, no caso, atrasados. Da mesma forma, os operadores cruzados avançados se tornam atrasados nas bordas B3 e B4, como pode ser visto no canto C4 com o uso do nó $[k-1, l-1]$.

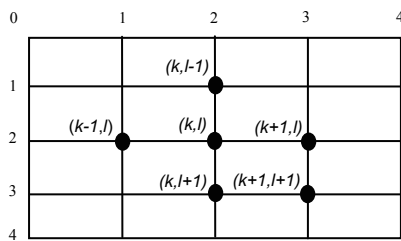


Figura 8: Antiga convenção para a derivada segunda.

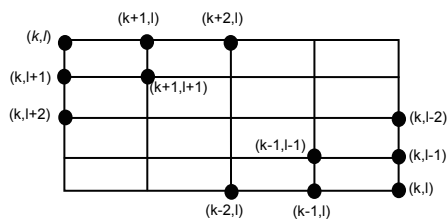


Figura 9: Nova convenção para a derivada segunda.

Uma maneira simples de pensar nessa discretização é que sempre são pegos pontos em direção ao centro da superfície, e nunca ao seu exterior. Além disso, para evitar forças internas muito grandes na fronteira somente a parcela \hat{E} da equação (13) é usada nas bordas. Dessa forma, os problemas relacionados às bordas, como instabilidade e falta de precisão, são atenuados, conforme os resultados apresentados na próxima seção.

5. Resultados

Como aqui estamos interessados na dinâmica convincente de tecidos, e não em uma determinada forma que um tecido pode assumir, todas as imagens são quadros de animações geradas que alcançam o equilíbrio físico. Tais animações são padronizadas em 400 quadros (1 a cada 50 de 20000 iterações), com um passo de tempo de 0,001, e podem ser visualizadas na página do projeto DeSMo já citada. Estes quadros serão discutidos na seção 5.1 Além disso, todos os dados de tempo apresentados na seção 5.2 foram medidos em um processador AMD de 3GHz equipado com uma placa de vídeo GeForce 6600GT.

5.1 Dinâmicas Realistas

Nesta implementação também é usado um arquivo de entrada com três grupos de parâmetros, relativos aos dados de superfície, de ambiente e de simulação. Os dados de superfície referem-se ao tipo de objeto, material de que é feito e resolução espacial que aproxima o tecido, enquanto os de ambiente fornecem as forças externas do meio no qual ele se encontra, como a força de gravidade e de fluido, que são exercidas em cada ponto. Os dados de simulação

correspondem aos parâmetros de tempo, como os dados apresentados na introdução desta seção, e também as restrições que são definidas sobre os nós da malha. Na figura 1(a), por exemplo, a simulação usa uma grade de 20x20 nós, definida no plano xy com 10 unidades de comprimento em y e 15 em x (parâmetros de superfície); força gravitacional de módulo e sentido $y = -9.8$ (parâmetros de ambiente); e ainda, dois pontos fixos (restrições nos cantos C1 e C2) para simular um mastro (parâmetros de simulação).

Cada restrição sobre a malha possui um intervalo de duração especificado pelas iterações, sendo possível simular colisões com diversos objetos sem ter nenhum tipo de tratamento para colisões. A poltrona da figura 1(b) foi feita desta forma, apenas definindo-se diferentes intervalos para os pontos fixos que representam a parte de cima de seu encosto (que são fixos com 10% das iterações processadas) e para os que representam seus braços (que são fixos com 20% das iterações processadas). Além disso, é possível também definir forças pontuais sobre a malha para simular os pequenos ajustes que ocorrem no mundo real, como empurrões em um pedaço de pano para amassá-lo, como mostra a figura 1(c). As figuras 10(a) e 10(b) ilustram as restrições usadas para fazer as figuras 1(b) e 1(c), respectivamente, considerando um círculo como ponto fixo e uma seta como força pontual.

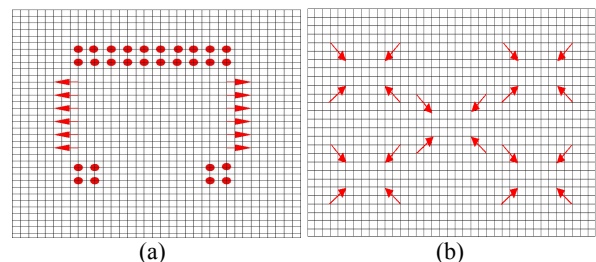


Figura 10: Restrições nas imagens (a) 1(b), (b) 1(c).

Na figura 10 as resoluções espaciais são meramente ilustrativas, não correspondendo ao tamanho das grades usadas efetivamente. Naturalmente, se terá uma maior precisão quanto maior for a resolução ao custo de um maior processamento. Entretanto, enquanto na bandeira da imagem 1(a) foram usados 400 vértices para produzir um resultado realista, uma mesma simulação desta bandeira, porém com apenas 64 vértices, é exibida na figura 11 para mostrar que nosso modelo consegue manter efeitos visuais convincentes para um número reduzido de nós, o que é muito interessante para os jogos.



Figura 11: Bandeira aproximada por uma grade 8x8.

Contudo, quando o realismo estiver em primeiro plano, como em uma cena pré-processada de um jogo, é possível elevar consideravelmente o número de vértices e, mesmo assim, ter um tempo razoável, como é mostrado na Tabela 1. A figura 12 mostra uma cortina de 10000 vértices, na qual foram inseridas apenas restrições de forças pontuais em duas linhas da grade para simular o trilha de uma cortina.

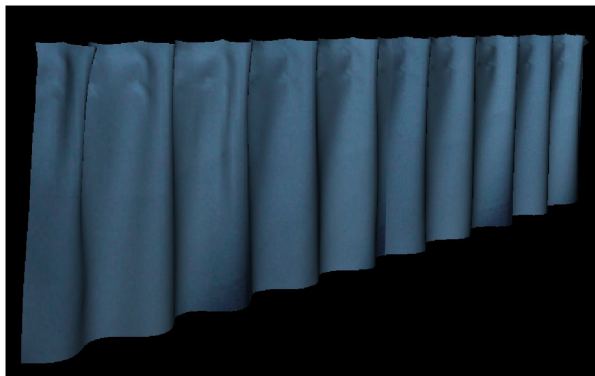


Figura 12: Cortina de 10000 vértices.

Por fim, as novas condições de fronteira resultam em um melhor efeito visual nas bordas, conforme mostrado na figura 13, sendo (a) uma imagem real e (b) a imagem simulada.

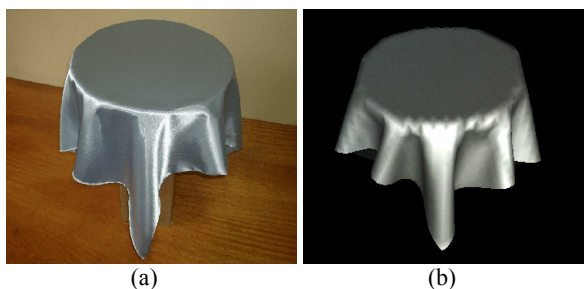


Figura 13: Comparação com fotografia.

5.2 Análise de Complexidade

Além das dinâmicas realistas, nosso modelo ainda conseguiu um bom desempenho nos tempos das simulações. O gráfico da figura 14 demonstra que nosso esquema de integração explícita executa as simulações em um tempo linear, de acordo com o número de vértices da superfície.

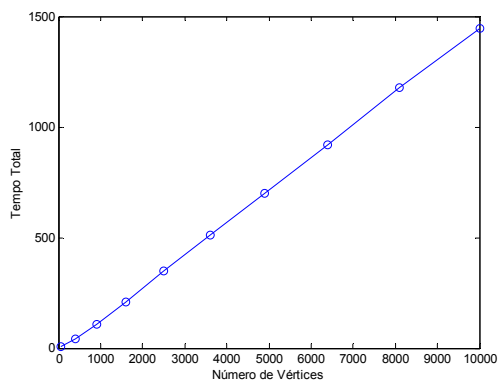


Figura 14: Complexidade linear.

O tempo total desse gráfico refere-se ao montante gasto para processar as 20000 iterações de uma simulação. A tabela 1 apresenta os dados correspondentes às simulações das imagens usadas neste artigo, sendo T o tempo total e t o tempo gasto por iteração, ambos especificados em segundos.

Tabela 1: Tempos por iteração.

Figuras	Número de Vértices	T	t
1a	400	41	$2,05^{-3}$
1b	2500	350	$1,75^{-2}$
1c	3600	511	$2,55^{-2}$
11	64	6	$3,00^{-4}$
12	10000	1446	$7,23^{-2}$
13	1600	207	$1,03^{-2}$

Dessa forma, é possível reparar que o tempo gasto em cada iteração é relativamente pequeno de acordo com a quantidade dos vértices utilizados, que estão diretamente ligados ao realismo do tecido. Na bandeira, por exemplo, o tempo gasto por iteração para encontrar as novas posições dos 400 vértices é muito pequeno, permitindo que várias iterações possam ser encontradas em uma parcela reduzida de um único passo de tempo da simulação em tempo real.

6. Conclusão

Apesar de ainda não termos uma simulação em tempo real, os tempos por iteração apresentados neste trabalho mostram que os esforços em perseguir esta linha de pesquisa são promissores. Além disso, nossa proposta não possui nenhum tipo de algoritmo de detecção de colisões, pois visa apenas propor uma nova forma de alcançar uma simulação de tecidos mais realista que também seja eficiente, o que não invalida usá-la como base e sobre ela desenvolver técnicas relativas à colisão. O acréscimo de algoritmos que realizem o tratamento de colisões e autocolisões, bem com a implementação deste modelo na GPU, poderia tornar nosso modelo aplicável a diversas simulações em tempo real, como os jogos.

Como trabalhos futuros, pensamos em modificar a parametrização do domínio da superfície, estender o modelo para malhas triangulares irregulares, analisar com maior profundidade a integração Verlet e buscar uma melhor estimativa para os elementos da geometria diferencial. Na implementação corrente os espaçamentos Δx^1 e Δx^2 pertencem ao intervalo $(0,1)$, o que faz com que a malha acabe sendo limitada não pelo método de integração, mas pelos espaçamentos do domínio, que não podem ser muito pequenos devido a problemas numéricos. Além disso, essa implementação possui a limitação de utilizar somente uma triangulação regular da malha, sendo possível obter, com a inclusão de triangulações irregulares, um realismo ainda maior. Quanto à integração Verlet, que foi a última parte implementada neste trabalho, ainda é preciso otimizá-la e adaptá-la para ser melhor comparada com a integração Euler. Por fim, existem diversas formas, além da utilizada neste modelo, para estimar os

parâmetros geométricos da geometria diferencial sobre uma superfície discreta. O tensor métrico, o tensor de curvatura, o vetor normal e os vetores de base, por exemplo, são elementos fundamentais no modelo e amplamente discutidos na literatura, portanto, uma comparação entre as diferentes formas de calculá-los seria interessante para descobrir qual é a melhor estimativa.

Agradecimentos

O primeiro autor gostaria de agradecer a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo suporte financeiro durante o período de 2004-2006 dentro do programa de Mestrado da Unicamp. A segunda autora gostaria de agradecer a Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) pelo apoio financeiro através da concessão de verba para o Projeto Individual 2003/13090-6.

Referências

BARAFF, D., E WITKIN, A., 1998. Large steps in cloth simulation. In: *Proceedings of SIGGRAPH'98*, 43-54.

BARAFF, D., WITKIN, A., E KASS, M. 2003. Untangling cloth. In: *ACM Transactions on Graphics* 22, 3 July, 862-870.

BARGMANN, R., 2003. *Real-time cloth simulation*. PhD thesis, Swiss Federal Institute of Technology Zurich .

BOXERMAN, E., 2003. *Speeding up cloth simulation*. MSc thesis, University of McGill.

BREEN, D.E., HOUSE D.H., E WOZNY, M.J., 1994. . A particle-based model for simulating the draping behavior of woven cloth. *Textile Research Journal*, November 1994, 64(11), 663-685.

BRIDSON, R., FEDKIW, R., E ANDERSON, J., 2002. Robust treatment of collisions, contact and friction for cloth animation. In: *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, 594-603.

BRIDSON, R., MARINO, S., E FEDKIW, R., 2003. Simulation of clothing with folds and wrinkles. In: *Eurographics / SIGGRAPH Symposium on Computer Animation*.

CORDIER, F. E THALMANN N.M., 2002. Real-time Animation of Dressed Virtual Humans. In: *EUROGRAPHICS 2003*, 21(3).

DESBRUN, M., SCHRÖDER, P., E BARR, A., 1999. Interactive animation of structured deformable objects. In: *Proceedings of Graphics Interface'99, June 1999 Ontario*, 1-8.

EISCHEN, J., DENG, S. E CLAPP, T., 1996. Finite-element modeling and control of flexible fabric parts. *IEEE Computer Graphics and Application* 16, 5, 71-80.

FEYNMAN, C.R., 1986. *Modeling the appearance of cloth*. MSc thesis, MIT, Cambridge.

GREEN, A.E., NAGHDI, P. E WAINWRIGHT, W., 1965. J., DENG, S. E CLAPP, T., 1996. A general theory of a cosserat surface. *Archive for Rational Mechanics and Analysis*, 20, 287-308.

HAUMANN, D.R., 1987. Modeling the physical behavior of flexible objects. *ACM SIGGRAPH'87 Course Notes #17 – Topics in Physically-Based Modeling, July 1987 Anaheim*.

JAKOBSEN, T., 2001. Advanced Character Physics. Disponível em: <http://www.teknikus.dk/tj/gdc2001.htm> [acessado em 15 Out 2006].

KOLB, A. E CUNTZ, N., 2005. Dynamic particle coupling for GPU-based fluid simulation. Disponível em: http://www.cg.informatik.uni-siegen.de/data/Publications/2005/asim_fluid_simulation_paper.pdf [acessado em 20 Ago 2006].

MELO, V.F., 2004. *Modelagem e Controle de Caimento e Dobras em Superfícies Deformáveis*. Tese de Doutorado, Universidade Estadual de Campinas.

OSHITA, M. E MAKINOCHI, A., 2001. Real-time Cloth Simulation with Sparse Particles and Curved Faces. In: *Proceedings of Computer Animation, November 2001 Seoul*.

PROVOT, X., 1995. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In: *Proceedings of Graphics Interface, 1995*, 147-154.

SIMO, J.C. E FOX, D.D., 1989. On a stress resultant geometrically exact shell model – Part I: formulation and optimal parameterization. *Computer Methods in Applied Mechanics and Engineering*, 72, 267-304.

SMITH, G. D., 1985. *Numerical solution of partial differential equations: finite difference methods*. New York: Oxford University Press.

TERZOPOULOS, D., PLATT, J., BARR, A., E FLEISCHER, K., 1987. Elastically Deformable Models. *Computer Graphics (Proceedings of SIGGRAPH) July 1987*, 21(4), 205-214.

VOLINO, P. E THALMANN, N. M., 1999. Fast geometric wrinkles on animated surfaces. In: *WSCG'99, Feb 1999 Plzen*.

WU, S.-T., E MELO, V.F., 2004. A Deformable Surface Model on the basis of the Theory of a Cosserat Surface. In: *Proceedings of SIBGRAPI 2004*.

WU, S.-T., E MELO, V.F., 2003. An approximation for normal vectors of deformable models. In: *Proceedings of SIBGRAPI 2003*, 3-10.