

Estimativa de propriedades geométricas de superfícies representadas em malhas de topologia arbitrária para aplicações em animações de tecidos

Matias Costa , Wu, Shin - Ting (Orientadora)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, 13083-970 – Campinas, SP, Brasil

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

Resumo – The usual techniques used for cloth animation have some limitations. The models used to represent the surfaces, using spring-mass systems, do not explicitly deal with bending behavior of the cloth. We are investigating an alternative paradigm on the bases of the Cosserat surface, which requires the first and second fundamental forms of the surface. Using the works of Rusinkiewicz and Batagelo among others as a starting point, an algorithm was developed for estimating those geometric quantities at the samples of the surface discretized as a mesh of arbitrary topology. Since our objective is, in a sequence of this work, to produce real-time cloth animations, the performance of the algorithm, besides its precision, is a serious issue. Several evaluation tests have been performed.

Palavras-chave – geometric properties, arbitrary topology meshes, cloth animation

1. Introdução

A técnica mais difundida para animações de tecidos é a utilização de um sistema massa-mola para simular o comportamento físico da superfície ao decorrer do tempo. Embora possua a vantagem de usar sistemas conhecidos e onde é possível realizar os cálculos rapidamente, esta abordagem sofre sérias limitações além de exigir diversos ajustes como, por exemplo, fatores atenuantes para evitar uma animação com muitas oscilações. Um dos motivos é que este modelo não possui uma forte relação com as características físicas do tecido.

Uma forma alternativa de representar tecidos é utilizando propriedades de geometria diferencial, através das quais é possível descrever o comportamento do curvamento de forma direta. Utilizando essas propriedades, Green et al[3] propõem um modelo, baseado na teoria de superfícies de Cosserat, para se calcular as energias presentes e forças aplicadas, de forma a gerar uma simulação mais próxima do comportamento físico.

O foco desse trabalho até o momento é o cálculo dessas propriedades de geometria diferencial, em específico a primeira e segunda formas fundamentais de superfícies, visando aplicar a teoria de superfícies de Cosserat para gerar animações. Nessa etapa, está sendo feita uma avaliação dos erros, utilizando algumas superfícies de controle, além de já levar em consideração a eficiência do algoritmo, pois é desejável que as animações possam ser gera-

das em tempo real.

2. Fundamentos

Descrevemos a seguir de forma abreviada algumas propriedades de geometria diferencial que nos interessam. Uma elaboração detalhada pode ser encontrada em [2]. Sendo $\mathbf{r}(u,v)$ uma superfície no espaço cartesiano e $\alpha(t) = r(u(t), v(t))$ uma curva sobre ela, a primeira forma fundamental para um ponto que pertence a \mathbf{r} e $\alpha(t)$, e possui um vetor normal $\mathbf{n}(t)$ é dada por

$$IFF(\alpha(t)) = \begin{bmatrix} \frac{du}{dt} & \frac{dv}{dt} \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \frac{du}{dt} \\ \frac{dv}{dt} \end{bmatrix},$$

onde

$$\begin{aligned} a_{11} &= \frac{\partial \mathbf{r}}{\partial u} \frac{\partial \mathbf{r}}{\partial u} & a_{12} &= \frac{\partial \mathbf{r}}{\partial u} \frac{\partial \mathbf{r}}{\partial v} \\ a_{21} &= \frac{\partial \mathbf{r}}{\partial v} \frac{\partial \mathbf{r}}{\partial u} & a_{22} &= \frac{\partial \mathbf{r}}{\partial v} \frac{\partial \mathbf{r}}{\partial v} \end{aligned} \quad (1)$$

A segunda forma fundamental é dada por

$$IIF(d^2\alpha(t), \mathbf{n}) = \begin{bmatrix} \frac{du}{dt} & \frac{dv}{dt} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} \frac{du}{dt} \\ \frac{dv}{dt} \end{bmatrix},$$

onde

$$\begin{aligned} b_{11} &= \frac{\partial^2 \mathbf{r}}{\partial u \partial u} \cdot \mathbf{n} & b_{12} &= \frac{\partial^2 \mathbf{r}}{\partial u \partial v} \cdot \mathbf{n} \\ b_{21} &= \frac{\partial^2 \mathbf{r}}{\partial v \partial u} \cdot \mathbf{n} & b_{22} &= \frac{\partial^2 \mathbf{r}}{\partial v \partial v} \cdot \mathbf{n} \end{aligned} \quad (2)$$

A matriz de coeficientes a_{ij} , onde $i,j=(1,2)$, é chamada de tensor métrico, pois dá uma medida da área da superfície em torno do ponto e a de b_{ij} é chamada de tensor de curvatura e, através de medida da variação do vetor normal, nos dá um indicativo de como a curvatura da superfície varia com as direções. Em condições onde a base desses tensores, no caso $\{\frac{\partial r}{\partial u}, \frac{\partial r}{\partial v}\}$ é ortonormal, o tensor métrico fica igual ao que chamamos de matriz de Weingarten (w_{ij}), que possui duas propriedades que nos interessam: (1) Os auto-valores representam as curvatura principais da superfície para o ponto; (2) Os auto-vetores representam, na base de referencia, as direções principais para o ponto.

3. Proposta

Nossa proposta para o cálculo desses tensores, tem como objetivo seguir as seguintes metas:

- **Erros pequenos:** Como os valores serão utilizados como base para uma outra seqüência de cálculos, erros presentes aqui irão se propagar por toda a simulação. É fundamental que eles sejam os menores possíveis.
- **Topologia arbitrária:** Nosso objetivo é poder produzir modelos mais fieis às características físicas dos tecidos. Para tal a técnica desenvolvida precisa ser aplicável em malhas com qualquer topologia.
- **Rapidez:** Para que seja possível simulações em tempo real, é preciso que o algoritmo seja eficiente e de fácil paralelização.
- **Comparação de valores:** Como o intuito é trabalhar com animações, é preciso poder comparar valores obtidos entre instantes diferentes, quando a malha tem formas distintas.

O trabalho desenvolvido por Batagelo e Wu[1], já tem um enfoque nos três primeiros pontos. Uma das diferenças entre o trabalho dele e o de Rusinkiewicz[4] foi considerar apenas a vizinhança "1-anel" do vértice sendo processado. Enquanto Rusinkiewicz processava cada face ligada ao ponto, Batagelo e Wu olhavam apenas os vértices diretamente conectados. Foi constatado que os resultados possuíam um erro um pouco maior, mas o tempo de processamento do tensor da curvatura era reduzido pela metade. A nossa proposta aqui é então estender

esse trabalho, tentando minimizar esses erros, manter a eficiência, adicionar o cálculo do tensor métrico e permitir comparações do mesmo ponto em instantes distintos da simulação. Este último ponto é importante não só devido às nossas metas, como também para poder avaliar se os valores calculados estão corretos. Rusinkiewicz já observa que isso é dado no algoritmo proposto por ele, por uma aproximação onde primeiro se rotaciona as bases para que fiquem no mesmo plano, e então é feita uma projeção das componentes dos tensores. Isso introduz um erro proporcional a quão grande é a rotação dos planos. No cálculo das propriedades de geometria diferencial, isso corresponde a considerar como se todas as arestas de um vértice estivessem no mesmo plano, o que é uma aproximação razoável para vértices com curvaturas pequenas. Entretanto para curvaturas mais acentuadas, ou mesmo para a nossa necessidade de comparar em instantes diferentes da animação, a rotação necessária passa a ser muito grande, e conseqüentemente o erro é considerável.

A alternativa que propusermos foi utilizar a relação existente quando se quer converter um tensor de uma base $\{u, v\}$ para uma $\{\tilde{u}, \tilde{v}\}$. Esta relação é

$$\begin{aligned} \tilde{a}_{11} &= a_{11} \frac{\partial u}{\partial \tilde{u}} \frac{\partial u}{\partial \tilde{u}} + a_{12} \frac{\partial u}{\partial \tilde{u}} \frac{\partial v}{\partial \tilde{u}} \\ &\quad + a_{21} \frac{\partial v}{\partial \tilde{u}} \frac{\partial u}{\partial \tilde{u}} + a_{22} \frac{\partial v}{\partial \tilde{u}} \frac{\partial v}{\partial \tilde{u}} \\ \tilde{a}_{12} = \tilde{a}_{21} &= a_{11} \frac{\partial u}{\partial \tilde{u}} \frac{\partial u}{\partial \tilde{v}} + a_{12} \frac{\partial u}{\partial \tilde{u}} \frac{\partial v}{\partial \tilde{v}} \\ &\quad + a_{21} \frac{\partial v}{\partial \tilde{u}} \frac{\partial u}{\partial \tilde{v}} + a_{22} \frac{\partial v}{\partial \tilde{u}} \frac{\partial v}{\partial \tilde{v}} \\ \tilde{a}_{22} &= a_{11} \frac{\partial u}{\partial \tilde{v}} \frac{\partial u}{\partial \tilde{v}} + a_{12} \frac{\partial u}{\partial \tilde{v}} \frac{\partial v}{\partial \tilde{v}} \\ &\quad + a_{21} \frac{\partial v}{\partial \tilde{v}} \frac{\partial u}{\partial \tilde{v}} + a_{22} \frac{\partial v}{\partial \tilde{v}} \frac{\partial v}{\partial \tilde{v}} \end{aligned} \quad (3)$$

Embora a notação esteja indicando tensor métrico, a relação é válida também para o tensor de curvatura. Os termos $\frac{\partial u}{\partial \tilde{u}}, \frac{\partial v}{\partial \tilde{u}}, \frac{\partial u}{\partial \tilde{v}}$ e $\frac{\partial v}{\partial \tilde{v}}$ podem ser obtidos se olharmos a matriz Jacobiana de transformação de coordenadas entre as bases. Com isso a conversão e comparação de tensores deixa de usar a aproximação de assumir que todos os planos são iguais.

3.1. Algoritmo

O algoritmo, de maneira semelhante ao implementado por Batagelo e Wu, estima a curvatura para

	$E_m a_{11}$	$E_m a_{12}$	$E_m a_{22}$	$E_m b_{11}$		$E_m b_{12}$		$E_m b_{22}$	
	C. W	C. W	Co. Wu	C. W.	Russ.	C. W.	Russ.	C. W.	Russ.
$\mathbf{h}(u,v)$	0.00281	0.00014	0.00277	0.01622	0.03925	0.00706	0.00685	0.00948	0.02112
$\mathbf{e}(u,v)$	0.01452	0.00324	0.01412	0.26362	0.65936	0.08002	0.08893	0.17422	0.37398

Tabela 1. Erros médios absolutos na estimativa dos tensores para $\mathbf{h}(u,v)$ e $\mathbf{e}(u,v)$

cada uma das arestas conectadas ao vértice sendo processado.

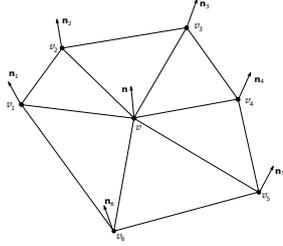


Figura 1. Vizinhança de um vértice da malha

Quando processando um vértice v como indicado na Figura 1, é escolhida uma das arestas como v_0v

$$\begin{aligned} \mathbf{t} &= \frac{v_0v}{|v_0v|} \\ \mathbf{b} &= \mathbf{n} \times \mathbf{t} \\ \mathbf{n} &= \mathbf{n} \end{aligned}$$

A variação da normal ao longo de cada aresta contribui para o cálculo e formação de um sistema que nos fornece a matriz de Weingarten, cujos auto-vetores indicam as direções principais na base $\{\mathbf{t}, \mathbf{b}, \mathbf{n}\}$. A escolha dessa base ortonormal implica que essa matriz seja igual ao tensor de curvatura, e o tensor métrico tenha a forma de uma matriz identidade.

O algoritmo como um todo portanto tem, até esse momento duas fases: (1) O cálculo das normais de todos os vértices da malha, feito pela forma tradicional de uma média das normais das superfícies vizinhas; (2) O cálculo dos tensores de curvatura e métrico para todos os vértices.

4. Resultados

Para poder se fazer uma avaliação dos erros foram adotadas duas superfícies cujas as parametrizações são conhecidas, $\mathbf{h}(u,v) = (u, v, (u^2 - v^2))$ e $\mathbf{e}(u,v) = (u, v, (\sin(2u) \cdot \cos(2v)))$. Portanto, é possível calcular analiticamente os seus tensores utilizando Eq. 1 e 2. A Figura 2 ilustra as formas

dessas superfícies, que foram escolhidas por possuírem certos tipos de pontos críticos que queríamos avaliar.

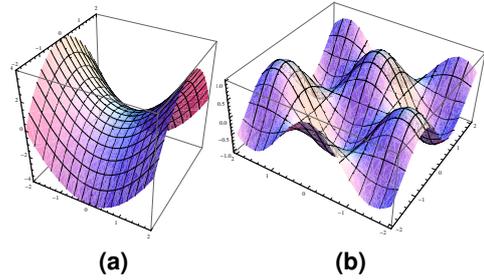


Figura 2. Superfícies de controle: a) $\mathbf{h}(u,v)$, b) $\mathbf{e}(u,v)$

Com isso podemos avaliar os erros médios dos valores estimados em relação aos valores obtidos analiticamente. Na Tabela 1 sintetizamos os erros médios absolutos, chamados de E_m , comparando os do nosso algoritmo, Costa Wu (C.W.), com os do algoritmo de Rusinkiewicz. Observe que só há uma coluna para cada componente do tensor métrico pois o método de Rusinkiewicz não os incluiu nos cálculos.

As Figuras 3 e 4 apresentam uma forma para visualizar comparativamente os erros em cada amostra: mapeá-los em cores. A cor verde representa um erro próximo a zero, enquanto que a cor azul ocorre em um limiar escolhido para avaliar os erros. Finalmente, a cor vermelha indica os maiores erros encontrados.

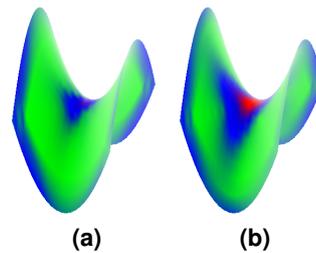


Figura 3. Erros na estimativa do tensor de curvatura, com limiar (azul) igual a 0.05. (a) Costa Wu (b) Rusinkiewicz.

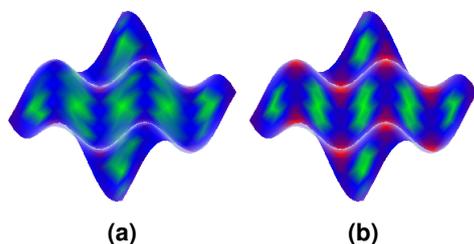


Figura 4. Erros na estimação do tensor de curvatura, com limiar (azul) igual a 0.1. (a) Costa Wu (b) Rusinkiewicz.

Os erros em $\mathbf{e}(u,v)$ são mais expressivos nos dois algoritmos em relação a $\mathbf{h}(u,v)$, mas isso era esperado já que a superfície possui diversos pontos onde a curvatura é acentuada. Na comparação entre nosso algoritmo e a proposta de Rusinkiewicz, é constatar uma maior precisão no nosso. Avaliamos ainda o desempenho de ambas as técnicas, com o resultado expresso na Figura 5.

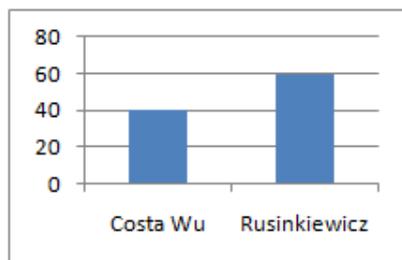


Figura 5. Tempo (em ms) dos dois algoritmos

5. Conclusões

O nosso algoritmo produz resultados significativamente melhores que a técnica de Rusinkiewicz em um tempo 33% menor, e ainda com o acréscimo de calcular o tensor métrico. A proposta de Batagelo e Wu, na parte específica de cálculo de tensores de curvatura, era mais eficiente, tendo medidas de tempo com aproximadamente metade do tempo das de Rusinkiewicz, entretanto com resultados numéricos piores.

Embora nossas medidas de tempo ainda não possibilitem fazer uma simulação em tempo real, especialmente porque ainda é preciso implementar a simulação física baseada na teoria de superfície de Cosserat, existe ampla oportunidade para melhorar o tempo de processamento através de paralelização, já que o processamento de cada vértice só depende

do estado anterior dos vértices vizinhos e portanto pode ser feito de maneira independente. Ao mesmo tempo, como mostrado por Batagelo e Wu[1], é possível implementar esse processamento em GPU e melhorar o desempenho ainda mais.

O trabalho prossegue na implementação para se utilizar a teoria de superfícies de Cosserat em animações de tecido. Os resultados obtidos até o momento mostram que a nossa implementação é boa e está produzindo estimativas de tensores com erros menores dos que os presentes em Batagelo e Wu [1] e Rusinkiewicz [4] o que nos deixa confiante em utilizar estes valores para as próximas etapas.

Referências

- [1] Batagelo, H. C.; Wu, S. T.; Estimating Curvatures and Their Derivatives on Meshes of Arbitrary Topology from Sampling Directions, *The Visual Computer*, 23:9-11, pp. 803-812, Springer, June 2007.
- [2] Carmo, M. P. *Geometria Diferencial de Curvas e Superfícies*, 2005. ISBN: 85-85818-26-3
- [3] A. E. Green, P.M. Naghdi, e W.L. Wainwright. A general theory of a cosserat surface. *Archive for Rational Mechanics and Analysis*, 20:287-308, 1965
- [4] Szymon Rusinkiewicz. Estimating Curvatures and Their Derivatives on Triangle Meshes. *Symposium on 3D Data Processing, Visualization, and Transmission*, Sep/2004