

Campos de Distância

Waldemar Celes

celes@inf.puc-rio.br

Tecgraf, DI/PUC-Rio

12 de Outubro de 2011

Campo de Distância 3D

Campo de distância é uma representação onde, para cada ponto do campo, sabe-se sua distância em relação ao ponto mais próximo do objeto.

Aplicações:

Em geral, usado como técnica de aceleração.

- ▶ Modelagem
- ▶ Manipulação/edição
- ▶ Visualização
- ▶ Detecção de colisão

Campo de distância contínuo

Campo de distância sem sinal

Considerando o conjunto Σ :

$$\text{dist}_{\Sigma}(\mathbf{p}) = \inf_{\mathbf{x} \in \Sigma} \|\mathbf{x} - \mathbf{p}\|$$

Campo de distância com sinal

Considerando o sólido S , a função de distância com sinal retorna a distância à fronteira ∂S :

$$d_S(\mathbf{p}) = \text{sign}(\mathbf{p}) \inf_{\mathbf{x} \in \partial S} \|\mathbf{x} - \mathbf{p}\|$$

$$\text{sign}(\mathbf{p}) = \begin{cases} -1 & \text{if } \mathbf{p} \in S \\ 1 & \text{otherwise} \end{cases}$$

Gradiente do campo de distância

Iso-superfície

É um conjunto de pontos: $\mathbf{p} / d(\mathbf{p}) = \tau$, onde τ é o iso-valor.

Gradiente

Para campo de distância com sinal, tem-se a seguinte propriedade:

$$\|\nabla d\| = 1$$

- ▶ Com exceção em pontos equidistantes à múltiplos pontos da superfície (e.g. centro de uma esfera)
 - ▶ Nestes pontos, o gradiente não é definido
 - ▶ Nos demais pontos, o gradiente é ortogonal à iso-superfície que passa no ponto

Derivada de segunda ordem

Provê informações sobre a curvatura das iso-superfícies.

Matrix de Hessian:

$$H = \begin{pmatrix} d_{xx} & d_{xy} & d_{xz} \\ d_{yx} & d_{yy} & d_{yz} \\ d_{zx} & d_{zy} & d_{zz} \end{pmatrix}$$

- ▶ Curvatura média:

$$k_M = \frac{1}{2}(d_{xx} + d_{yy} + d_{zz})$$

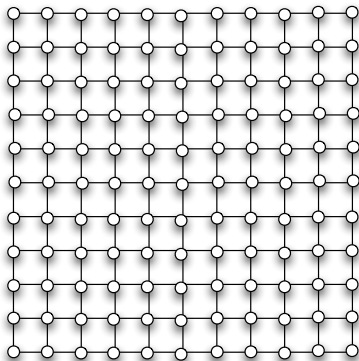
- ▶ Curvatura Gaussiana:

$$k_G = \begin{vmatrix} d_{xx} & d_{xy} \\ d_{yx} & d_{yy} \end{vmatrix} + \begin{vmatrix} d_{xx} & d_{xz} \\ d_{zx} & d_{zz} \end{vmatrix} + \begin{vmatrix} d_{yy} & d_{yz} \\ d_{zy} & d_{zz} \end{vmatrix}$$

- ▶ As curvaturas principais são dadas pelos dois auto-vetores diferentes de zero de H
 - ▶ O outro auto-valor é zero, significando que d varia linearmente ao longo do gradiente

Campo de distância discreto

Obtido por uma amostragem de um campo de distância contínuo.



Reconstrução

Em geral, estimada através de núcleos de reconstrução finitos.

Construção de campos de distância

Algoritmo força bruta:

- ▶ Para cada voxel (amostra), calcula a menor distância ao objeto
- ▶ Na prática, inviável devido ao alto custo computacional

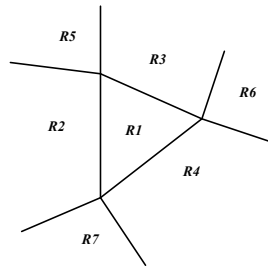
Objetos representados por malhas de triângulos:

- ▶ Malhas devem ser fechadas, orientáveis e *2-manifold*
 - ▶ Malha não contém auto-interseção
 - ▶ Cada aresta tem exatamente dois triângulos adjacentes
 - ▶ Triângulos adjacentes a um vértice devem formar um único ciclo ao redor do vértice
- ▶ Algoritmo força-bruta: $O(N.M)$

Campo de distância para malhas de triângulos

Distância entre ponto e triângulo:

- ▶ Projeção do ponto no plano do triângulo
 - ▶ Região $R1$
 - ▶ Distância do ponto ao plano
 - ▶ Regiões $R2, R3, R4$
 - ▶ Distância do ponto à reta correspondente
 - ▶ Regiões $R5, R6, R7$
 - ▶ Distância do ponto ao ponto (vértice) correspondente



Campo de distância para malhas de triângulos

Técnicas de aceleração

- ▶ Reduzir triângulos testados explorando coerência espacial
 - ▶ Uso de hierarquias: $O(N.M) \rightarrow O(N.\log(M))$
- ▶ Calcular apenas distâncias de voxel próximos e propagar para demais voxels
 - ▶ Uso de transformadas de distância

Uso de hierarquia

Hierarquia de caixas envolventes (AABB)

- ▶ Para cada triângulo, calcula a caixa envolvente
- ▶ Triângulos são agrupados por coerência espacial, formando uma hierarquia de caixas envolventes

Cálculo da distância a um ponto

- ▶ Para cada caixa visitada, calcula d_{min} e d_{max}
- ▶ O valor d_{min} é usado como chave de uma lista de prioridade
- ▶ Caixas com d_{min} maior que um d_{max} de outra caixa, podem ser descartadas

Distância de voxels próximos à superfície

Inversão do processo:

- ▶ Para cada triângulo, calcula AABB considerando o critério de proximidade
- ▶ Calcula a distância para todos os voxels dentro da AABB
- ▶ Guarda o valor mínimo de cada voxel

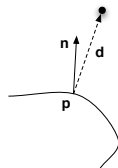
Um método mais sofisticado identifica "características"

- ▶ Identificam ponto mais próximo de sub-espacos
- ▶ Malha é transformada em poliedros (*as características*)
 - ▶ Faces se transformam em prismas
 - ▶ Arestas se transformam em cunhas
 - ▶ Vértices se transformam em pirâmides com base poligonal

Determinação do sinal

Superfície com continuidade C^1

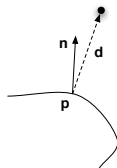
$$\mathbf{n} \cdot \mathbf{d} = \begin{cases} > 0 & \text{outside} \\ < 0 & \text{inside} \end{cases}$$



Determinação do sinal

Superfície com continuidade C^1

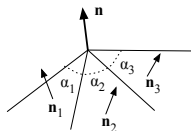
$$\mathbf{n} \cdot \mathbf{d} = \begin{cases} > 0 & \text{outside} \\ < 0 & \text{inside} \end{cases}$$



Malhas de triângulos não são C^1

- ▶ Determinação com traçado de raio
- ▶ Uso de pseudo-normais
 - ▶ Normal de vértices
 - ▶ Média das normais dos triângulos incidentes ponderada pelos ângulos

$$\mathbf{n} = \frac{\sum_i \mathbf{n}_i \alpha_i}{\|\sum_i \mathbf{n}_i \alpha_i\|}$$



- ▶ Normal de arestas
 - ▶ Extensão da mesma ideia: ângulos iguais a π

Transformada de distância

Chamfer distance transformation

Forward pass

	f		f		
f	e	d	e	f	
	d		d		
f	e	d	e	f	
	f		f		

$z=-2$

f	e	d	e	f
e	c	b	c	e
d	b	a	b	d
e	c	b	c	e
f	e	d	e	f

$z=-1$

	d		d	
d	b	a	b	d
	a	0		

$z=0$

Backward pass

		0	a	
d	b	a	b	d
	d		d	

$z=0$

f	e	d	e	f
e	c	b	c	e
d	b	a	b	d
e	c	b	c	e
f	e	d	e	f

$z=1$

	f		f	
f	e	d	e	f
	d		d	
f	e	d	e	f
	f		f	

$z=2$

Values Used to Produce Each Chamfer Type

Transform	a	b	c	d	e	f
City Block (Manhattan)	1					
Chessboard	1	1				
Quasi-Euclidean $3 \times 3 \times 3$	1	$\sqrt{2}$				
Complete Euclidean $3 \times 3 \times 3$	1	$\sqrt{2}$	$\sqrt{3}$			
$\langle a, b, c \rangle_{opt} 3 \times 3 \times 3 [102]$	0.92644	1.34065	1.65849			
Quasi-Euclidean $5 \times 5 \times 5$	1	$\sqrt{2}$	$\sqrt{3}$	$\sqrt{5}$	$\sqrt{6}$	3

/* Forward Pass */

FOR($z = 0$; $z < f_z$; $z++$)

FOR($y = 0$; $y < f_y$; $y++$)

FOR($x = 0$; $x < f_x$; $x++$)

$F[x, y, z] =$

$\inf_{\forall i, j, k \in fp} (F[x+i, y+j, z+k] + m[i, j, k])$

/* Backward Pass */

FOR($z = f_z - 1$; $z \geq 0$; $z--$)

FOR($y = f_y - 1$; $y \geq 0$; $y--$)

FOR($x = f_x - 1$; $x \geq 0$; $x--$)

$F[x, y, z] =$

$\inf_{\forall i, j, k \in bp} (F[x+i, y+j, z+k] + m[i, j, k])$

Referência

Jones, M., Baerentzen, J., and Sramek, M. (2006). 3d distance fields: a survey of techniques and applications. *Visualization and Computer Graphics, IEEE Transactions on*, 12(4):581–599.