Mathematische Sicht auf Computergraphik

Laplace-Operatoren auf polygonalen Meshes

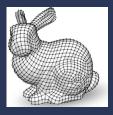
Antonia Stavemann 09.02.2023

TechTalk TU Berlin

freitagsrunde.org/
TechTalk



ZIEL: Operatoren wie den Laplacian auf polygonalen Meshes zu definieren



[Ebke et al. 2014]



[Ebke et al. 2014]

ZIEL: Operatoren wie den Laplacian auf polygonalen Meshes zu definieren



[Ebke et al. 2014]



[Ebke et al. 2014]

Wir nehmen uns die Angst vor der Mathematik in der Computergrafik

Wir nehmen uns die Angst vor der Mathematik in der Computergrafik - am Beispiel von dem Laplace Operator auf polygonalen Meshes



[Ebke et al. 2014]



[Ebke et al. 2014]

COMPUTERGRAFIK

- computergestützte Darstellung von 2-dim Kurven und 3-dim Flächen
- nicht nur Einzelbilder, sondern auch Animationen (Rauch, Wellen, ...)
- durch Funktionen/Operatoren auf einem Mesh

Mesh

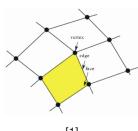
KOMBINATORISCHES SET UP

Mesh

00000000000000000

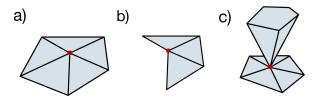
Polygonal Mesh $\mathcal{P} = (V, E, F)$

- V Menge von Vertices
- E Menge an Edges $e := (v_i, v_{i+1}) \in E$
- F Menge an Faces $f = (v_1, ..., v_{n_f})$



MANNIGFALTIGKEIT

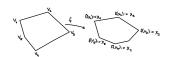
• alle Edges formen einen offenen oder geschlossenen Fan



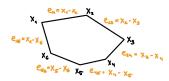
00000000000000000

Mesh

- $f: V \to \mathbb{R}^3$ mit $f(v_i) \neq f(v_i)$
- Vertex Positions Matrix $\mathbf{X}_f = [\mathbf{x}_1...\mathbf{x}_{n_f}]^T$



• Edge Positions $\mathbf{e}_{ij} = \mathbf{x}_i - \mathbf{x}_j$

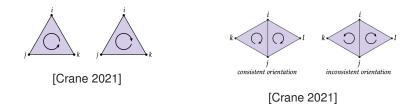


ORIENTIERUNG

000000000000000000

Mesh

- können jede Edge in zwei Halfedges unterteilen
- · orientierbar: wir finden eine einheitliche globale Orientierung

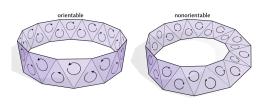


ORIENTIERUNG

00000000000000000

Mesh

- können jede Edge in zwei Halfedges unterteilen
- · orientierbar: wir finden eine einheitliche globale Orientierung



[Sharp 2021]

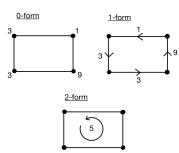
SKALARFUNKTIONEN AUF MESHES

- können Funktionen auf unser Mesh anwenden
- glatte Skalarfunktion: $u : \mathbb{R}^n \to \mathbb{R}$
- 0-Form: $\alpha:V\to\mathbb{R}$

Mesh

000000000000000000

- 1-Form: $\alpha: E \to \mathbb{R}$
- 2-Form: $\alpha: F \to \mathbb{R}$



Mesh

Differenz Operator

$$\mathbf{D}_f = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & -1 \end{bmatrix}$$

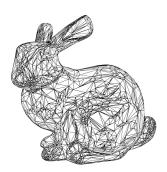
• Durchschnittsoperator
$$\mathbf{A}_f = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0 & 0.5 \end{bmatrix}$$

Edge-Mittelpunkte

$$\mathbf{B}_f = \mathbf{A}_f \mathbf{X}_f$$

DREIECKSMESHES:

- 3 Punkte im Raum definieren eine Ebene eindeutig
- · können Normale definieren
- · Operatoren sind gut definiert



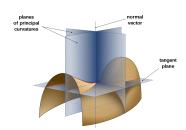
[Daniel Kaufmann]

Wieso brauchen wir dann Polygone?

VIERECKSMESH:

MATHEMATISCHE VORTEILE:

• zwei Hauptkrümmungen



[Gaba 2006]

VIERECKSMESH:

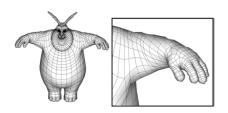
VORTEILE IN GEOMETRY PROCESSING:

- Modellierungs- und Ingenieursoftware beruht auf Polygonen
- genauere Ergebnisse, da Polygone nicht eben sein müssen
- Polygone sehen natürlicher aus

VIERECKSMESH:

VORTEILE IN GEOMETRY PROCESSING:

- Modellierungs- und Ingenieursoftware beruht auf Polygonen
- genauere Ergebnisse, da Polygone nicht eben sein müssen
- · Polygone sehen natürlicher aus
- intuitiver zu manipulieren



[Bommes et al. 2012]

Wieso reichen uns keine Vierecke?

POLYGONALE MESH:

Euler-Charakteristik:

$$\chi = F + V - E = 2 \cdot (1 - g)$$

F: Anzahl der Faces

E: Anzahl der Edges

V: Anzahl der Vertices

g: Anzahl der Löcher



[Bommes et al. 2012]

POLYGONALE MESH:

Euler-Charakteristik:

$$\chi = F + V - E = 2 \cdot (1 - g)$$

F: Anzahl der Faces

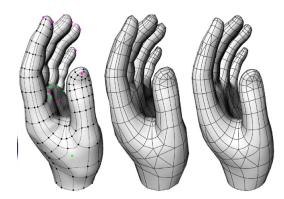
E : Anzahl der Edges

V: Anzahl der Vertices

g: Genus / Anzahl der Hänkel

Wir können ein reguläres Vierecksmesh nicht auf jede beliebige Figur legen.

POLYGONAL MESH:

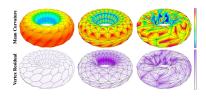


[Alliez et al. 2003]

Wieso triangulieren wir die Polygone nicht?

TRIANGULATION

- · Polygone werden in mehrere ebene Dreiecke unterteilt
- führt zu höheren Abweichungen



[De Goes et al. 2020]

erstes Bild: unser Ziel zweites/drittes Bild: trianguliertes Mesh

Laplace Operator

Wieso der Laplace-Operator?

LAPLACE-BELTRAMI OPERATOR OVERVIEW

Geometrie:

- Krümmung
- Invarianz unter Isometrie

Physik:

- Wärmeleitungsgleichung / Heat Equation
- Wellengleichung / Wave Equation







$$\frac{d^2}{dt^2}u = \Delta u$$

[Crane 2021]

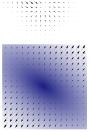
LAPLACE-BELTRAMI OPERATOR OVERVIEW

Divergenz:

- Wie weit streben die Vektoren in einer sehr kleinen Umgebung des Punktes auseinander?
- in Senken: negative Divergenz
- · ist sie gleich Null: quellenfrei

Gradient:

- In welcher Richtung haben wir den größten Anstieg?
- zeigt in die Richtung der größten Änderung



[Crane 2021]

LAPLACE OPERATOR IM \mathbb{R}^n

- Skalarfunktion $u: \mathbb{R}^n \to \mathbb{R}$
- $\Delta u = \sum_{i=1}^{n} \frac{\partial^2}{\partial x_i^2} u$

Beispiel:

Mesh

$$u(x,y) = -x^2 - 2y^2$$

$$\Delta u = \frac{\partial^2}{\partial x \partial x} (-x^2 - 2y^2) + \frac{\partial^2}{\partial y \partial y} (-x^2 - 2y^2)$$

$$= \frac{\partial}{\partial x} (-2x) + \frac{\partial}{\partial y} (-4y)$$

$$= -2 - 4 = -6$$

Kann dargestellt werden als $\Delta u = div(grad(u))$

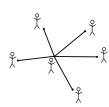
GRAPH LAPLACIAN

Beispiel:

Mesh

- Freundesgruppe mit der Anzahl der Leistungspunkte
- Laplacian: Hast du im Durchschnitt mehr oder weniger Leistungspunkte als deine Freunde?
- Laplacian gibt uns die Abweichung vom lokalen Durchschnitt

$$(Lu)_i = (\frac{1}{deg(i)} \sum_{ij \in E} u_j) - u_i$$



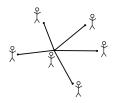
GRAPH LAPLACIAN

Beispiel:

Mesh

- Freundesgruppe mit der Anzahl der Leistungspunkte
- Laplacian: Hast du im Durchschnitt mehr oder weniger Leistungspunkte als deine Freunde?
- Laplacian gibt uns die Abweichung vom lokalen Durchschnitt

$$(Lu)_i = (\frac{1}{deg(i)} \sum_{ij \in E} u_j) - u_i$$



Wir betrachten bisher nur die Kombinatorik und nicht die Geometrie.

Aber wie kriegen wir diesen Laplace Operator nun auf Polygone?

NORMALE

- für Dreiecke einfach zu berechnen: Kreuzprodukt der Edge Vektoren
- · Polygone: Fläche unbekannt
- können nicht direkt die Normale auf dem Polygon berechnen
- brauchen Normale für den Laplacian

FLÄCHENINHALT

Mesh

Dreiecke

$$a_f = \frac{1}{2}wh$$

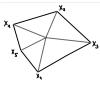
ebene Polygone

$$a_f = \frac{1}{2} \sum_{v_i \in f} x_i \times x_{i+1}$$

nicht-ebene Polygone

$$a_f = \frac{1}{2} \sum_{v_i \in f} \mathbf{x}_i \times \mathbf{x}_{i+1}$$







NORMALE

• die stetige Flächennormale ist equivalent ist zu:

$$Normale = \frac{Flächeninhalt}{|Flächeninhalt|}$$

Wieso brauchen wir verschiedene Ansätze?

EIGENSCHAFTEN

- lokal (wir betrachten nur die direkten Nachbarn)
- für konstante Funktionen: $\Delta u = 0$
- · positiv semidefinit
- konvergiert gegen den glatten Laplace Operator
- ..

EIGENSCHAFTEN

- lokal (wir betrachten nur die direkten Nachbarn)
- für konstante Funktionen: $\Delta u = 0$
- · positiv semidefinit
- · konvergiert gegen den glatten Laplace Operator
- ...
 - keiner der folgenden Ansätze erfüllt alle Eigenschaften
 - es muss abgewogen werden, welcher Ansatz für das Problem passend ist

1. Ansatz

DIE IDEE

Bekannte Zusammenhänge ausnutzen.

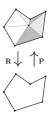
- Zusammenhang zwischen dem Gradienten des Flächeninhalts und dem Laplacian
- Gradient des Flächeninhalts: In welche Richtung müssen wir einen einzelnen Punkt bewegen, damit der Flächeninhalt größer wird?

 $\nabla_{x_i} | \text{Flächeninhalt} | = (\text{Laplace Operator * Vertex Positions}_{\text{pro face}})_i$

2. Ansatz

IDEE

- durch Triangulierung verlieren wir Eigenschaften
- aber nicht, wenn wir einen virtuellen Vertex in der Mitte platzieren
- Idee:
 - 1. Fügen einen virtuellen Vertex ein
 - 2. Erhalten ein Dreiecksnetz
 - 2. Können Methoden für Dreiecke anwenden

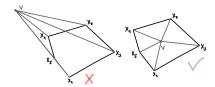


[Bunge 2020]

VIRTUELLER VERTEX

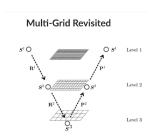
Mesh

- · einfach zu berechnen und eindeutig
- der Flächeninhalt soll sich nicht vergrößern, sondern nur verkleinern
- für ebene Polygone: im Inneren des Polygons
- darstellbar als Kombination der anderen Vertices des Faces. (Affine Kombination)



DIE GALERKIN MULTI-GRID METHOD

- gegeben: zwei Meshes, die die gleiche Oberfläche approximieren
- · hier: polygones Mesh und Mesh mit virtuellem Vertex
- · Operator auf dem kleineren Mesh anwenden



[Verdugo, Wall 2016]

DIE GALERKIN MULTI-GRID METHOD

Prolongation Matrix:

Mesh

$$\begin{bmatrix}
 x_1 \\
 x_2 \\
 x_3 \\
 x_4 \\
 \sum_{i=1}^4 \omega_i x_i
 \end{bmatrix} = \begin{bmatrix} 1 \\
 1 \\
 1 \\
 \omega_1 \omega_2 \omega_3 \omega_4
 \end{bmatrix} \begin{bmatrix} x_1 \\
 x_2 \\
 x_3 \\
 x_4
 \end{bmatrix}$$

Restriction Operator:

• transponierte Prolongation Matrix: $R = P^T$

Somit können wir die Matrizen von dem Laplacian auf Dreiecksmeshes "sandwichen": $S_{Polyaone} = P^T S_{Dreiecke} P$

3. Ansatz

3. ANSATZ

Mesh

• es gilt: $\Delta u = div(grad(u))$

Hauptziel: Diskretisierung des Gradienten

GRADIENT AUF DREIECKSMESHES

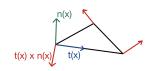
Satz von Stokes:

$$\int_{f} (u(x)) dx = \oint_{\partial f} u(x) \underbrace{t(x) \times n(x)}_{\text{Einheits normale}} dx$$

u(x): Skalarfunktion

n(x): Normale

t(x): Tangente an dem Vertex x



Wieso können wir das nicht genauso auf Polygone anwenden?

Wieso können wir das nicht genauso auf Polygone anwenden?

Das Normalenfeld auf nicht-ebenen Polygonen ist nicht definiert.

COGRADIENT - POLYGONAL MESHES

· Drehen die Gleichung auf beiden Seiten um 90 Grad

CoGradienten:

Mesh

$$\nabla^\perp u(x) := [n(x)]_\times \nabla u(x)$$

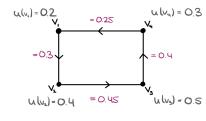
Satz von Stokes:

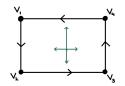
$$\int_{f} \nabla^{\perp} u(x) \, dx = \oint_{\partial f} u(x) t(x) \, dx$$

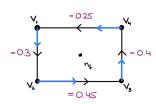
DISKRETER GRADIENT

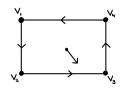
Mesh

$$G_f u_f = \sum_{v \in f} -\frac{1}{a_f} n_f \times (x_{v_{i+1}} - x_{v_i}) \frac{1}{2} (u_f(v_i) + u_f(v_{i+1}))$$









Zusammenfassung

ZUSAMMENFASSUNG

- · wir haben gelernt, wozu der Laplacian verwendet werden kann
- wir konnten verschiedene Laplacians diskretisieren auf ebenen und nicht-ebenen Polygonen
- · verschiedene Ansätze kennengelernt

Mich interessiert das Thema. Wo lerne ich mehr?

MODULE:

Informatikstudierende:

- · Computer Graphik 1
- · Computer Graphik 2

Mathematikstudierende:

- · Mathematische Visualisierung
- · Geometrie 3

Außerdem:

• Folien werden hochgeladen und Paper verlinkt, siehe freitagsrunde.org/techtalk