

Gittertypen und Interpolation

Andreas Kolb und Martin Lambers

computergraphik und multimedia systeme
universität siegen



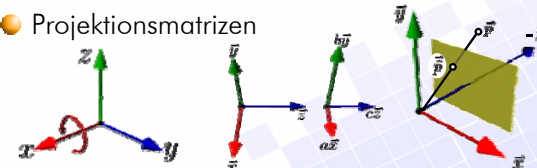
Review: Letzte Stunde

2

Wiederholung Lineare Algebra:

Homogene Koordinaten:

- erlauben Gleichbehandlung von Punkten (Ortsvektoren) und Richtungsvektoren
- Rotation, Translation und Skalierung (Affine Abbildungen) in einer Transformationsmatrix
- Projektionsmatrizen

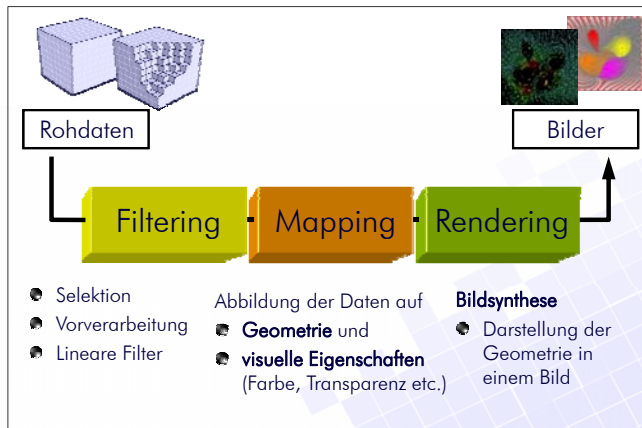


Andreas Kolb, Martin Lambers

Visualisierung

Review: Letzte Stunde

3

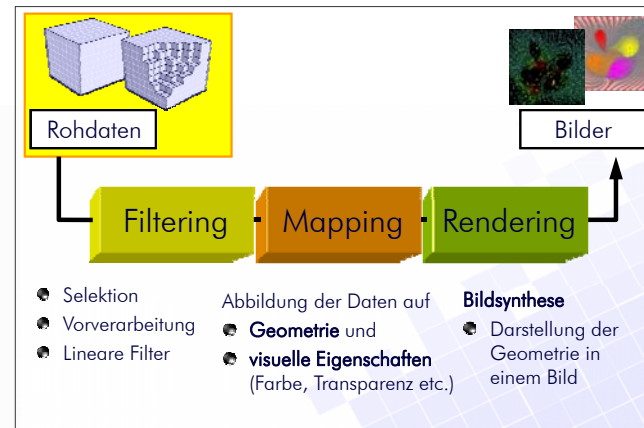


Andreas Kolb, Martin Lambers

Visualisierung

Review: Letzte Stunde

4



Andreas Kolb, Martin Lambers

Visualisierung

Heute: Gitterstrukturen

5

Wozu brauchen wir Gitter?

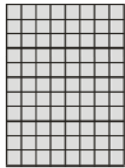
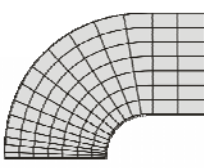
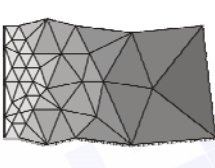
- Natürliche Phänomene sind in der Regel kontinuierlich (d.h. unendlich hoch aufgelöst)
- Computer können (natürlich) nur mit endlichen Datenmengen arbeiten
- Kontinuierliche Daten müssen **abgetastet** (digitalisiert, rasterisiert) werden, um im Computer verarbeitet werden zu können.

Andreas Kolb, Martin Lambers

Visualisierung

2D Gittertypen

6

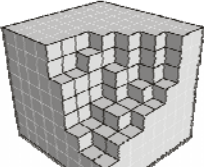
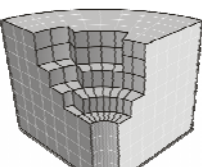
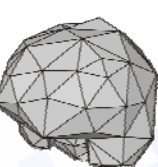
uniform rectilinear	curvilinear	unstrukturiert
		
Zellen: gleichförmige Rechtecke, Quadrate Anordnung: uniform regulär	Zellen: unterschiedliche Vierecke, Anordnung: nicht uniform, aber strukturiert (Topologie)	Zellen: Dreiecke, (selten Rechtecke, Polygone) Anordnung: unstrukturiert

Andreas Kolb, Martin Lambers

Visualisierung

3D Gittertypen

7

uniform rectilinear	curvilinear	unstrukturiert
		
Zellen: gleichförmige Hexaeder, Anordnung: uniform regulär	Zellen: unterschiedliche Hexaeder, Anordnung: nicht uniform, aber strukturiert	Zellen: Tetraeder, Hexaeder, Pyramiden, Prismen Anordnung: unstrukturiert

Andreas Kolb, Martin Lambers

Visualisierung

Allgemeine Gitterstrukturen

8

Speicherbedarf für Gitter

Was muß gespeichert werden?

Allgemein:

- Dimension
- Liste der Vertices (Gitterpunkte) [explizit oder implizit]
- Liste der Zellen [explizit oder implizit]

Optional (redundant):

- Liste der Kanten
- Liste der Flächen (3D)
- Kreuzreferenzen: Zelle → Vertices, Zelle → Flächen, Zelle → Kanten, Fläche → Kanten, usw.

Andreas Kolb, Martin Lambers

Visualisierung

Uniforme Gitter

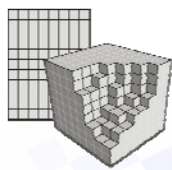
9

Strukturiertes Gitter

Was muß gespeichert werden?

Zellen und Vertices implizit:

- Dimension N_x, N_y, N_z :
Anzahl der Abtastpunkte in x-, y- und z-Richtung
- Zellgröße/Gitterweite $\Delta x, \Delta y, \Delta z$:
Abstand der Abtastpunkte in x-, y- und z-Richtung
- Datenwerte als Array mit Index n
 $n = N_x \cdot N_y \cdot z + N_x \cdot y + x$



Andreas Kolb, Martin Lambers

Visualisierung

Rectilineare Gitter

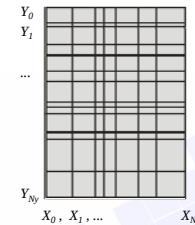
10

Strukturiertes Gitter

Was muß gespeichert werden?

Zellen implizit, Vertices explizit:

- Dimension N_x, N_y, N_z :
Anzahl der Abtastpunkte in x-, y- und z-Richtung
- Koordinaten der Linien/Ebenen:
 - $X_0, X_1, \dots, X_{N_x}$,
 - $Y_0, Y_1, \dots, Y_{N_y}$,
 - $Z_0, Z_1, \dots, Z_{N_z}$,
- Datenwerte als Array mit Index n
 $n = N_x \cdot N_y \cdot z + N_x \cdot y + x$



Andreas Kolb, Martin Lambers

Visualisierung

Curvilineare Gitter

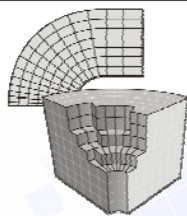
11

Strukturiertes Gitter

Was muß gespeichert werden?

Zellen implizit, Vertices explizit:

- Dimension N_x, N_y, N_z :
Anzahl der Abtastpunkte in x-, y- und z-Richtung
- Liste der Gitterpunkte P_{ijk} :
x-, y- und z-Koordinaten der Gitterpunkte
- Datenwerte als Array mit Index n
 $n = N_x \cdot N_y \cdot z + N_x \cdot y + x$



Andreas Kolb, Martin Lambers

Visualisierung

Unstrukturiertes Gitter

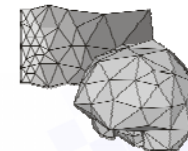
12

Unstrukturiertes Gitter

Was muß gespeichert werden?

Zellen und Vertices explizit:

- Anzahl der Gitterpunkte N
- Anzahl der Zellen Z
- Liste der Gitterpunkte P_{ijk} :
x-, y- und z-Koordinaten der Gitterpunkte
- Liste der Zellen:
z.B. Tetraeder: 4 Indices der Gitterpunkte



Andreas Kolb, Martin Lambers

Visualisierung

Zusammenfassung Gittertypen

13

Uniform rectilinear:

Vorteile: einfach, effizienter Speicherzugriff, Zellsuche ist trivial,

Nachteile: überall gleiche Auflösung, keine lokale Verfeinerung möglich

Curvilinear:

Vorteile: flexibler als rectilineare Gitter

Nachteile: Gitterstruktur muß zusätzlich gespeichert werden, Algorithmus zur Zellsuche erforderlich

Unstrukturiert:

Vorteile: sehr hohe Flexibilität, lokale Verfeinerung möglich

Nachteile: Position jedes Datenpunktes und Zellen müssen explizit gespeichert werden, Zellsuche erforderlich

Andreas Kolb, Martin Lambers

Visualisierung

Beispiel: Dreiecksnetz

14

„Winged Edge“

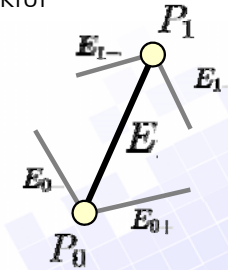
kantenbasierte Datenstruktur

Für jede Kante E :

2 Referenzen auf Vertices
4 Referenzen auf benachbarte Kanten

Für jeden Vertex P :

Position (3 Koordinaten)
1 Referenz auf eine seiner Kanten



Andreas Kolb, Martin Lambers

Visualisierung

Beispiel: Dreiecksnetz

15

„Shared Vertex“

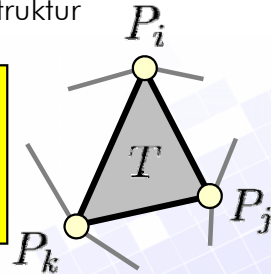
flächenbasierte Datenstruktur

Für jeden Vertex P :

Position (3 Koordinaten)

Für jedes Dreieck T :

3 Referenzen auf Vertices



Andreas Kolb, Martin Lambers

Visualisierung

Beispiel: Dreiecksnetz

16

Welche Datenstruktur ist effizienter?

Faustregel: Dreiecksnetz mit n Vertices, hat ca. $3n$ Kanten und $2n$ Flächen

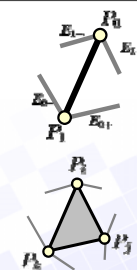
„Winged Edge“

benötigt ca. $(3+1)n + 6 \cdot 3n = 22n$ Werte

„Shared Vertex“

benötigt ca. $3n + 3 \cdot 2n = 9n$ Werte

ABER: Winged Edge ermöglicht schnellere Algorithmen, da die Traversierung (z.B. Aufzählung aller Kanten zu einem geg. Vertex, aller Vertices zu einem geg. Dreieck, etc.) in **konstanter** Zeit möglich ist!



Andreas Kolb, Martin Lambers

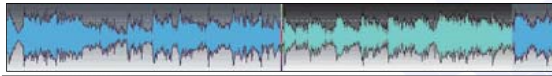
Visualisierung

Abtastung

17

Beispiel: Ein Audiosignal

- ist in der Natur eine Schallwelle, d.h. eine Schwingung eines Mediums (z.B. Luft)
- Im Rechner werden Audiosignale *gesampelt*, d.h. *abgetastet und quantisiert* (AD-Wandlung)



Abtasttheorem: Wie muß das Signal digitalisiert werden, damit bestimmte Eigenschaften erhalten bleiben?

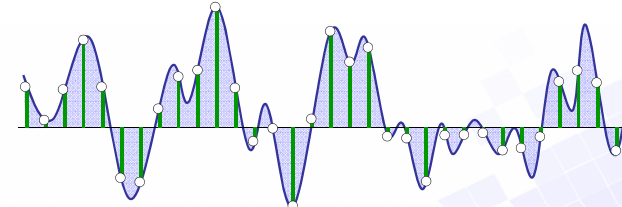
Andreas Kolb, Martin Lambers

Visualisierung

Diskrete Darstellung

18

Kontinuierliche Signale werden abgetastet,



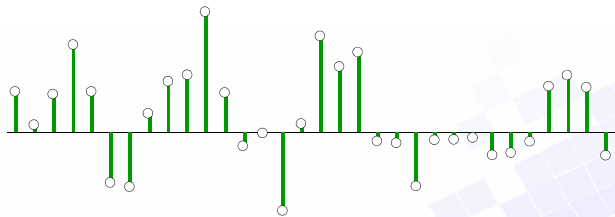
Andreas Kolb, Martin Lambers

Visualisierung

Diskrete Darstellung

19

Kontinuierliche Signale werden abgetastet,



- Das Ergebnis ist ein **diskretes** Signal, das aus einer Folge von Abtastwerten besteht.

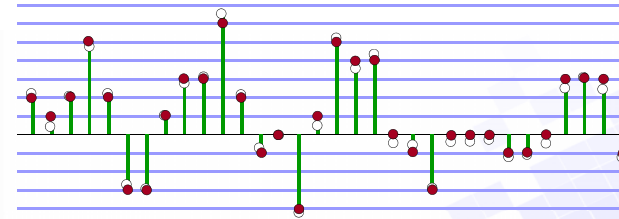
Andreas Kolb, Martin Lambers

Visualisierung

Quantisierung

20

Die diskreten Abtastwerte werden **quantisiert**,



... und mit einer bestimmten Genauigkeit (z.B. 32bit) abgespeichert.

Andreas Kolb, Martin Lambers

Visualisierung

Abtastung und Quantisierung ²¹

Der Fehler, der durch die Quantisierung entsteht, wird als **Quantisierungsrauschen** bezeichnet. (Wir vernachlässigen diesen Fehler im Folgenden)

Frage: Wie muß ich den Abstand der Abtastpunkte wählen (**Abtastfrequenz**) und welche Konsequenzen folgen daraus?

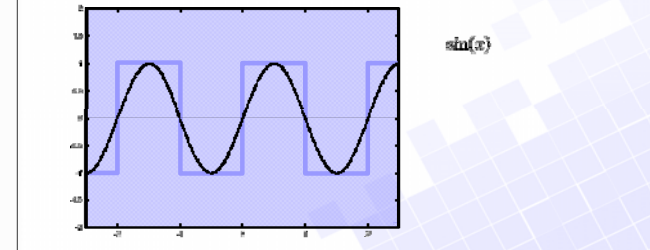


Andreas Kolb, Martin Lambers

Visualisierung

Frequenzspektrum ²²

Jedes kontinuierliche Signal $f(x)$ kann als Überlagerung von **harmonischen Schwingungen unterschiedlicher Frequenz** (Sinus und Cosinus) dargestellt werden.

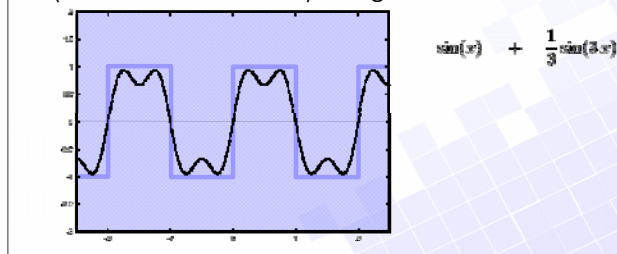


Andreas Kolb, Martin Lambers

Visualisierung

Frequenzspektrum ²³

Jedes kontinuierliche Signal $f(x)$ kann als Überlagerung von **harmonischen Schwingungen unterschiedlicher Frequenz** (Sinus und Cosinus) dargestellt werden.

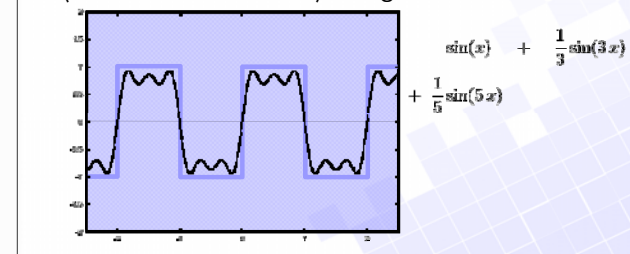


Andreas Kolb, Martin Lambers

Visualisierung

Frequenzspektrum ²⁴

Jedes kontinuierliche Signal $f(x)$ kann als Überlagerung von **harmonischen Schwingungen unterschiedlicher Frequenz** (Sinus und Cosinus) dargestellt werden.



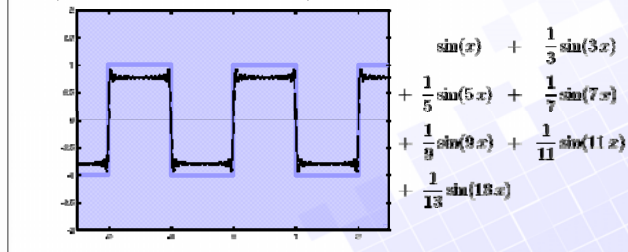
Andreas Kolb, Martin Lambers

Visualisierung

Frequenzspektrum

25

- Jedes kontinuierliche Signal $f(x)$ kann als Überlagerung von *harmonischen Schwingungen unterschiedlicher Frequenz* (Sinus und Cosinus) dargestellt werden.



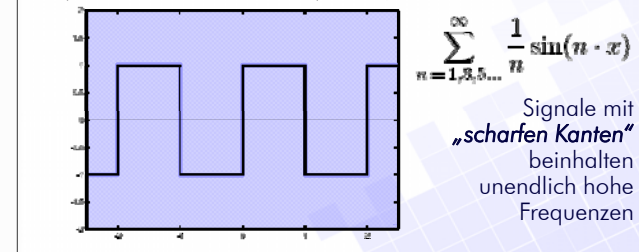
Andreas Kolb, Martin Lambers

Visualisierung

Frequenzspektrum

26

- Jedes kontinuierliche Signal $f(x)$ kann als Überlagerung von *harmonischen Schwingungen unterschiedlicher Frequenz* (Sinus und Cosinus) dargestellt werden.



Andreas Kolb, Martin Lambers

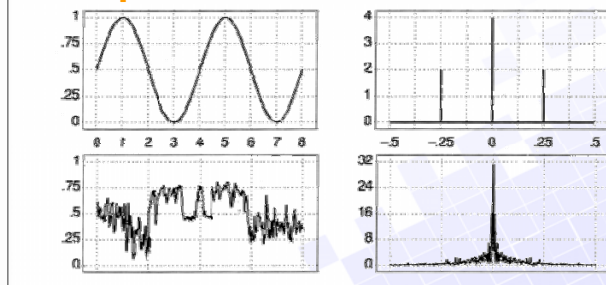
Visualisierung

Frequenzspektrum

27

- Jedes kontinuierliche Signal $f(x)$ hat ein eindeutiges *Frequenzspektrum* $F(t)$.

Beispiele:



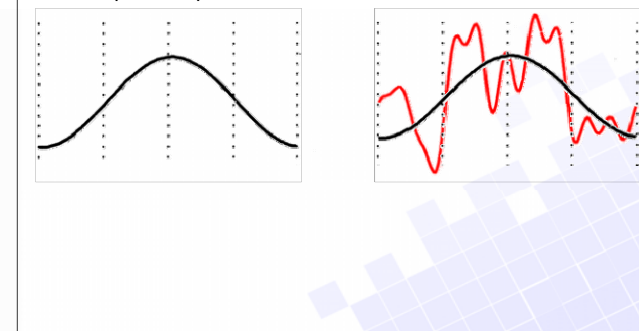
Andreas Kolb, Martin Lambers

Visualisierung

Frequenzspektrum

28

- Beispiel eines Signals mit beschränktem Frequenzspektrum:



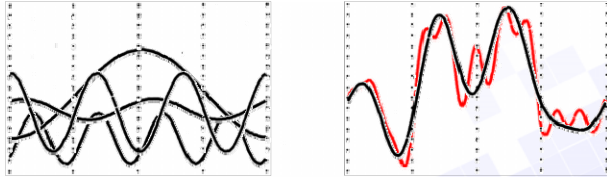
Andreas Kolb, Martin Lambers

Visualisierung

Frequenzspektrum

29

- Beispiel eines Signals mit beschränktem Frequenzspektrum:



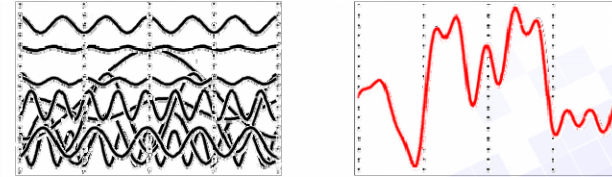
Andreas Kolb, Martin Lambers

Visualisierung

Frequenzspektrum

30

- Beispiel eines Signals mit beschränktem Frequenzspektrum:



- Ein Signal heißt **bandbegrenzt**, wenn sein Frequenzspektrum auf ein endliches Intervall beschränkt ist.

Andreas Kolb, Martin Lambers

Visualisierung

Abtasttheorem

31

- Ein kontinuierliches Signal $f(x)$, dessen Frequenzspektrum $F(t)$ bandbegrenzt ist, d.h.

$$F(t) = 0 \quad \text{für} \quad |t| > t_{\max}$$

läßt sich **exakt rekonstruieren** aus einer diskreten Abtastung mit einer Schrittweite:

$$\tau < \frac{1}{2t_{\max}}$$

Andreas Kolb, Martin Lambers

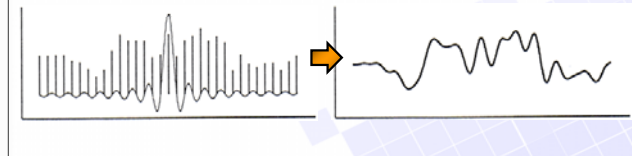
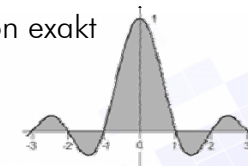
Visualisierung

Rekonstruktion

32

- Ein hinreichend genau abgetastetes Signal kann aus den diskreten Abtastwerten mit Hilfe der sinc-Funktion exakt rekonstruiert werden.

$$\text{sinc}(x) = \frac{\sin(x)}{x}$$



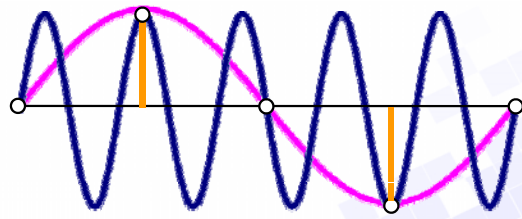
Andreas Kolb, Martin Lambers

Visualisierung

Abtasttheorem

33

Was passiert nun, wenn die Abtastfrequenz zu gering ist ?



Es wird ein falsches Signal rekonstruiert!

Aliasing

Andreas Kolb, Martin Lambers

Visualisierung

Literatur

34

Abtasttheorem nachzulesen in:

J. Foley, A. van Dam, S. Feiner, and J. Hughes.
Computer Graphics, Principle And Practice.
Addison-Wesley, 1993.

Chapter 14.10.3 Sampling Theory

oder in jedem Lehrbuch über Signalverarbeitung

Andreas Kolb, Martin Lambers

Visualisierung

Differenzieren

35

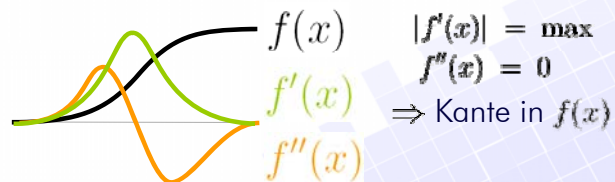
Ableitungen im 1D: $f'(x) = \frac{df(x)}{dx}$

Eigenschaften der Ableitung:

$f'(x)$ beschreibt die Steigung der Kurve $f(x)$

Lokale Extrema: $f(x) = \max \Rightarrow f'(x) = 0$

Kantendetektion:



Andreas Kolb, Martin Lambers

Visualisierung

Gradient

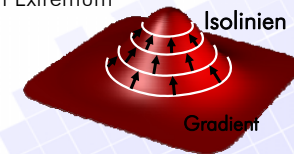
36

1. Ableitung im 2D:
Der Gradient $\text{grad} f(x, y) = \begin{pmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{pmatrix}$

Eigenschaften des Gradientenvektors:

- zeigt in die Richtung des größten Anstiegs
- steht senkrecht auf den Isolinen
- ist gleich Null bei lokalem Extremum

Beispiel:
2D Höhenfeld



Andreas Kolb, Martin Lambers

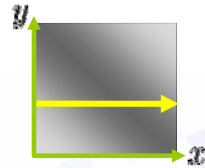
Visualisierung

Richtungsableitung

37

Partielle Ableitung im 2D:

● in x-Richtung: $\frac{\partial f(x,y)}{\partial x}$



Andreas Kolb, Martin Lambers

Visualisierung

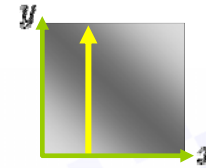
Richtungsableitung

38

Partielle Ableitung im 2D:

● in x-Richtung: $\frac{\partial f(x,y)}{\partial x}$

● in y-Richtung: $\frac{\partial f(x,y)}{\partial y}$



Andreas Kolb, Martin Lambers

Visualisierung

Richtungsableitung

39

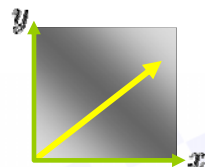
Partielle Ableitung im 2D:

● in x-Richtung: $\frac{\partial f(x,y)}{\partial x}$

● in y-Richtung: $\frac{\partial f(x,y)}{\partial y}$

● in beliebige Richtung $\vec{w} = \begin{pmatrix} w_x \\ w_y \end{pmatrix}$

$$\begin{aligned} \frac{\partial f(x,y)}{\partial \vec{w}} &= w_x \frac{\partial f(x,y)}{\partial x} + w_y \frac{\partial f(x,y)}{\partial y} \\ &= \langle \vec{w} \circ \text{grad} f(x,y) \rangle \end{aligned}$$



Andreas Kolb, Martin Lambers

Visualisierung

Ableitungen in 3D

40

● Gradientenvektor:

$$\text{grad} f(x,y,z) = \begin{pmatrix} \frac{\partial f(x,y,z)}{\partial x} \\ \frac{\partial f(x,y,z)}{\partial y} \\ \frac{\partial f(x,y,z)}{\partial z} \end{pmatrix}$$

● Richtungsableitung in Richtung $\vec{w} = \begin{pmatrix} w_x \\ w_y \\ w_z \end{pmatrix}$:

$$\begin{aligned} \frac{\partial f(x,y,z)}{\partial \vec{w}} &= w_x \frac{\partial f(x,y,z)}{\partial x} + w_y \frac{\partial f(x,y,z)}{\partial y} + w_z \frac{\partial f(x,y,z)}{\partial z} \\ &= \langle \vec{w} \circ \text{grad} f(x,y,z) \rangle \end{aligned}$$

Andreas Kolb, Martin Lambers

Visualisierung

Differenzieren auf Gittern

41

Wie kann ich Ableitungen berechnen auf diskreten Daten?

● **Möglichkeit 1:** Approximiere die diskreten Daten (lokal) durch eine differenzierbare Funktion und differenziere diese.

● **Möglichkeit 2:** Ersetze das Differential durch finite Differenzen

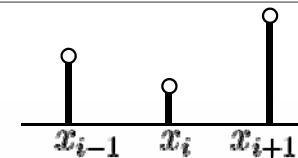
$$f'(x) = \frac{df(x)}{dx} \mapsto \frac{\Delta f(x)}{\Delta x}$$

Andreas Kolb, Martin Lambers

Visualisierung

Möglichkeit 1: Curve Fitting

42



Bestimme die Ableitung $f'(x_i)$ an der Stelle x_i

1. Finde ein Polynom 2. Grades,

$$f(x) = ax^2 + bx + c$$

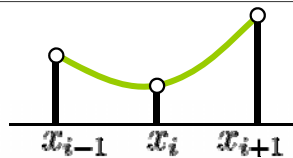
das die diskreten Werte lokal approximiert.

Andreas Kolb, Martin Lambers

Visualisierung

Möglichkeit 1: Curve Fitting

43



Bestimme die Ableitung $f'(x_i)$ an der Stelle x_i

1. Finde ein Polynom 2. Grades,

$$f(x) = ax^2 + bx + c$$

das die diskreten Werte lokal approximiert.

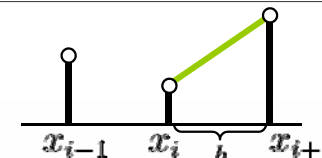
2. Differenziere dieses Polygon an der Stelle x_i

Andreas Kolb, Martin Lambers

Visualisierung

Möglichkeit 2: Finite Differenzen

44



Bestimme die Ableitung $f'(x_i)$ an der Stelle x_i

Vorwärtsdifferenzen:

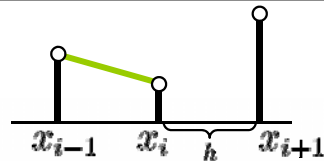
$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_i)}{h}$$

Andreas Kolb, Martin Lambers

Visualisierung

Möglichkeit 2: Finite Differenzen

45



Bestimme die
Ableitung $f'(x_i)$
an der Stelle x_i

Vorwärtsdifferenzen:

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_i)}{h}$$

Rückwärtsdifferenzen:

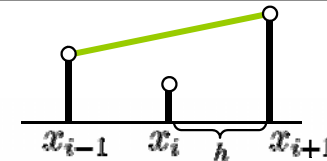
$$f'(x_i) \approx \frac{f(x_i) - f(x_{i-1}))}{h}$$

Andreas Kolb, Martin Lambers

Visualisierung

Möglichkeit 2: Finite Differenzen

46



Bestimme die
Ableitung $f'(x_i)$
an der Stelle x_i

Vorwärtsdifferenzen:

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_i)}{h}$$

Rückwärtsdifferenzen:

$$f'(x_i) \approx \frac{f(x_i) - f(x_{i-1}))}{h}$$

Zentrale Differenzen:

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_{i-1}))}{2h}$$

Fehlerordnung

 $o(h)$ $o(h)$ $o(h^2)$

Andreas Kolb, Martin Lambers

Visualisierung

Interpolation

47

Curve Fitting:

- Gegeben Werte $f(-1) = f_{-1}$, $f(0) = f_0$, $f(1) = f_1$
- Gesucht: Abschätzung der Ableitung/Steigung bei $x=0$
- Curve Fitting mit quadratischem Polynom
 - $f(x) = ax^2 + bx + c$; $f(1) - f(-1) = 2b$
 - Ableitung: $f'(x) = 2ax + b$
 - Konkret ergibt $f'(0) = b = \frac{1}{2}(f(1) - f(-1)) = \frac{f_1 - f_{-1}}{2}$
also dasselbe wie zentrale Differenzen

In nichts zeigt sich der Mangel an
mathematischer Bildung mehr, als in
einer übertrieben genauen Rechnung

(Carl Friedrich Gauss, dt. Mathematiker, 1777-1855)

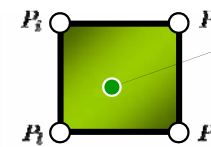
Andreas Kolb, Martin Lambers

Visualisierung

Lokale Interpolation

48

- An den Datenpunkten P_i sind Datenwerte gegeben.



Wie bestimme ich einen
Datenwert für einen Punkt,
der innerhalb einer Zelle
liegt?

Interpolation

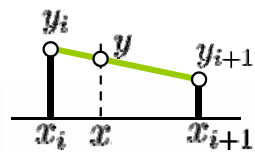
d.h. approximative Rekonstruktion des Signals

Andreas Kolb, Martin Lambers

Visualisierung

Lineare Interpolation (1D)

49



Lineares *Blenden*,

$$f(x) = \alpha y_{i+1} + (1 - \alpha) y_i$$

mit

$$\alpha = \frac{x - x_i}{x_{i+1} - x_i}$$

$$x = x_i \Rightarrow \alpha = 0 \Rightarrow f(x) = y_i$$

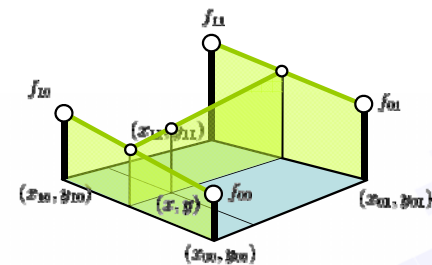
$$x = x_{i+1} \Rightarrow \alpha = 1 \Rightarrow f(x) = y_{i+1}$$

Andreas Kolb, Martin Lambers

Visualisierung

Bilineare Interpolation (2D)

50



$$f_0 = \alpha f_{10} + (1 - \alpha) f_{00}$$

$$f_1 = \alpha f_{11} + (1 - \alpha) f_{01}$$

$$f(x, y) = \beta f_1 + (1 - \beta) f_0$$

$$\alpha = \frac{x - x_{00}}{x_{10} - x_{00}}$$

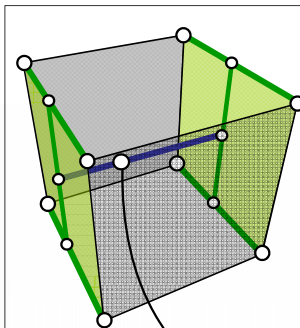
$$\beta = \frac{y - y_{00}}{y_{01} - y_{00}}$$

Andreas Kolb, Martin Lambers

Visualisierung

Trilineare Interpolation (3D)

51



4 lineare Interpolationen

$$f_{00} = \alpha f_{001} + (1 - \alpha) f_{000}$$

$$f_{01} = \alpha f_{011} + (1 - \alpha) f_{010}$$

$$f_{10} = \alpha f_{101} + (1 - \alpha) f_{100}$$

$$f_{11} = \alpha f_{111} + (1 - \alpha) f_{110}$$

2 bilineare Interpolationen

$$f_0 = \beta f_{01} + (1 - \beta) f_{00}$$

$$f_1 = \beta f_{11} + (1 - \beta) f_{10}$$

1 trilineare Interpolation

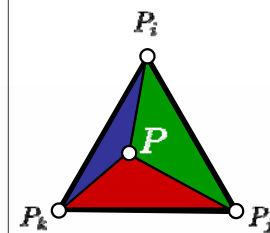
$$f = \gamma f_1 + (1 - \gamma) f_0$$

Andreas Kolb, Martin Lambers

Visualisierung

Dreieckige Zellen (2D)

52



Der Punkt P kann ausgedrückt werden als

$$P = a_i P_i + a_j P_j + a_k P_k$$

mit

$$a_i + a_j + a_k = 1$$

$$A = \text{Area}(\triangle P_i P_j P_k)$$

$$a_i = \frac{1}{A} \text{Area}(\triangle P P_j P_k)$$

$$a_j = \frac{1}{A} \text{Area}(\triangle P_i P P_k)$$

$$a_k = \frac{1}{A} \text{Area}(\triangle P_i P_j P)$$

Schwerpunktskoordinaten
(Baryzentrische Koordinaten)

$$a_i = a_j = a_k = \frac{1}{3}$$

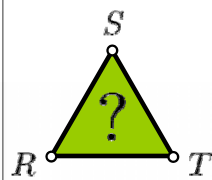
$\Rightarrow P$ ist Schwerpunkt

Andreas Kolb, Martin Lambers

Visualisierung

Dreieckige Zellen (2D)

53



Wie bestimme ich den
Flächeninhalt eines
Dreiecks $Area(\triangle RST)$?

$$R = \begin{pmatrix} R_x \\ R_y \end{pmatrix} \quad S = \begin{pmatrix} S_x \\ S_y \end{pmatrix} \quad T = \begin{pmatrix} T_x \\ T_y \end{pmatrix}$$

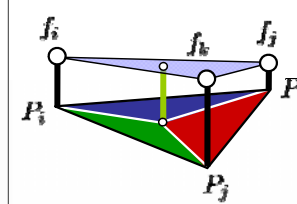
$$\begin{vmatrix} 1 & 1 & 1 \\ R_x & S_x & T_x \\ R_y & S_y & T_y \end{vmatrix} = \pm 2 \cdot Area(\triangle RST)$$

Andreas Kolb, Martin Lambers

Visualisierung

Lineare Interpolation im Dreieck

54



Bestimme die baryzentrischen
Koordinaten des Punkts P

$$P = a_i P_i + a_j P_j + a_k P_k$$

Bestimme den Funktionswert
mit den gleichen Gewichten

$$f = a_i f_i + a_j f_j + a_k f_k$$

Anmerkung: Auch Punkte P außerhalb des Dreiecks lassen
sich in baryzentrischen Koordinaten ausdrücken. Die
Gewichte sind dann nicht mehr alle positiv. Es gilt aber
nach wie vor $a_i + a_j + a_k = 1$

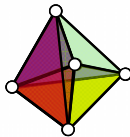
Andreas Kolb, Martin Lambers

Visualisierung

Tetraeder

55

Baryzentrische Koordinaten im Tetraeder



1. Verbinde den inneren Punkt
mit allen Randpunkten.

Es entstehen 4 Tetraeder als
Unterteilung

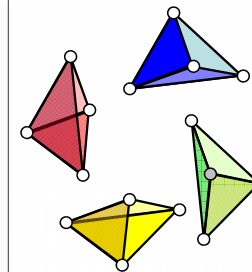
Andreas Kolb, Martin Lambers

Visualisierung

Tetraeder

56

Baryzentrische Koordinaten im Tetraeder



1. Verbinde den inneren Punkt
mit allen Randpunkten.

Es entstehen 4 Tetraeder als
Unterteilung

2. Das Volumen eines Teil-
Tetraeders ist das bary-
zentrische Gewicht für den
gegenüberliegenden Punkt

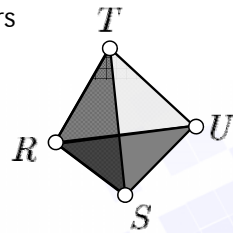
Andreas Kolb, Martin Lambers

Visualisierung

Tetraeder

57

● Volumen eines Tetraeders
 $\text{Vol}(\text{Tet}(R, S, T, U))$ mit
 Vertices R, S, T und U :



$$\begin{vmatrix} 1 & 1 & 1 & 1 \\ R_x & S_x & T_x & U_x \\ R_y & S_y & T_y & U_y \\ R_z & S_z & T_z & U_z \end{vmatrix} = \pm 6 \cdot \text{Vol}(\text{Tet}(R, S, T, U))$$

Andreas Kolb, Martin Lambers

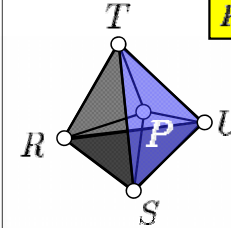
Visualisierung

Tetraeder

58

Baryzentrische Koordinaten im Tetraeder

$$P = a_R R + a_S S + a_T T + a_U U$$



$$a_R = \frac{\begin{vmatrix} 1 & 1 & 1 & 1 \\ P_x & S_x & T_x & U_x \\ P_y & S_y & T_y & U_y \\ P_z & S_z & T_z & U_z \end{vmatrix}}{\begin{vmatrix} 1 & 1 & 1 & 1 \\ R_x & S_x & T_x & U_x \\ R_y & S_y & T_y & U_y \\ R_z & S_z & T_z & U_z \end{vmatrix}}$$

Andreas Kolb, Martin Lambers

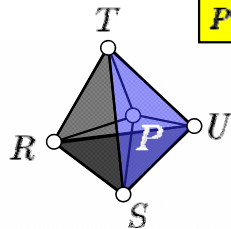
Visualisierung

Tetraeder

59

Baryzentrische Koordinaten im Tetraeder

$$P = a_R R + a_S S + a_T T + a_U U$$



$$a_S = \frac{\begin{vmatrix} 1 & 1 & 1 & 1 \\ R_x & P_x & T_x & U_x \\ R_y & P_y & T_y & U_y \\ R_z & P_z & T_z & U_z \end{vmatrix}}{\begin{vmatrix} 1 & 1 & 1 & 1 \\ R_x & S_x & T_x & U_x \\ R_y & S_y & T_y & U_y \\ R_z & S_z & T_z & U_z \end{vmatrix}}$$

Andreas Kolb, Martin Lambers

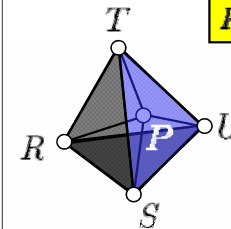
Visualisierung

Tetraeder

60

Baryzentrische Koordinaten im Tetraeder

$$P = a_R R + a_S S + a_T T + a_U U$$



$$a_T = \frac{\begin{vmatrix} 1 & 1 & 1 & 1 \\ R_x & S_x & P_x & U_x \\ R_y & S_y & P_y & U_y \\ R_z & S_z & P_z & U_z \end{vmatrix}}{\begin{vmatrix} 1 & 1 & 1 & 1 \\ R_x & S_x & T_x & U_x \\ R_y & S_y & T_y & U_y \\ R_z & S_z & T_z & U_z \end{vmatrix}}$$

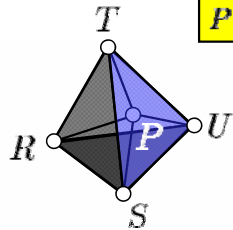
Andreas Kolb, Martin Lambers

Visualisierung

Tetraeder

61

Baryzentrische Koordinaten im Tetraeder



$$P = a_R R + a_S S + a_T T + a_U U$$

$$a_U = \frac{\begin{vmatrix} 1 & 1 & 1 & 1 \\ R_x & S_x & T_x & P_x \\ R_y & S_y & T_y & P_y \\ R_z & S_z & T_z & P_z \end{vmatrix}}{\begin{vmatrix} 1 & 1 & 1 & 1 \\ R_x & S_x & T_x & U_x \\ R_y & S_y & T_y & U_y \\ R_z & S_z & T_z & U_z \end{vmatrix}}$$

Interpolation:

$$f = a_R f_R + a_S f_S + a_T f_T + a_U f_U$$

Andreas Kolb, Martin Lambers

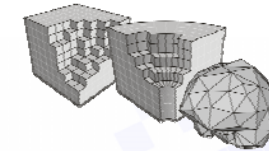
Visualisierung

Zusammenfassung

62

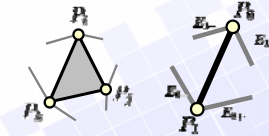
Gitterstrukturen in 2D und 3D

- uniforme Gitter
- rectilineare Gitter
- curvilineare Gitter
- unstrukturierte Gitter



Datenstrukturen für Dreiecksnetze

- „Shared Vertex“
- „Winged Edge“



Andreas Kolb, Martin Lambers

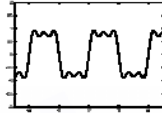
Visualisierung

Zusammenfassung

63

Abtasttheorem

- Kontinuierliche Signale lassen sich in harmonische Schwingungen zerlegen (Frequenzspektrum)
- Bandbegrenzte Signale lassen sich aus diskreten Abtastwerten exakt rekonstruieren
- Exakte Rekonstruktion erfolgt mit dem Sinc-Filter
- Unterabtastung führt zu **Aliasing**-Effekten



Andreas Kolb, Martin Lambers

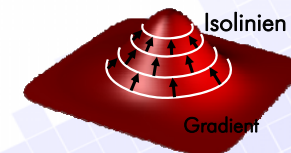
Visualisierung

Zusammenfassung

64

Differenzieren auf Gittern

- Gradientenvektor
 - zeigt in Richtung des größten Anstiegs
 - steht senkrecht auf den Isolinien (2D) bzw. Isoflächen (3D)
- Differenzieren auf Gittern
 - Curve/Surface Fitting
 - Finite Differenzen
 - Vorwärts, Rückwärts,
 - Zentrale Differenzen



Andreas Kolb, Martin Lambers

Visualisierung

Zusammenfassung

65

● Lokale Interpolation

- Stückweise lineare Interpolation
 - Kartesisch: Linear, bilinear, trilinear
 - Baryzentrisch: Dreieck und Tetraeder

Andreas Kolb, Martin Lambers

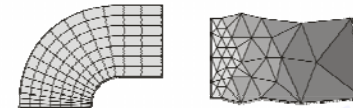
Visualisierung

Nächste Stunde

66

● Curvilineare und unstrukturierte Gitter

- Zellsuche
- Triangulierung/Tessellierung



● Filter

- Lineare und Nichtlineare Filter

Andreas Kolb, Martin Lambers

Visualisierung