

Marching Cubes

Corbinian Gruber

Computer Science Department
University of Freiburg

Albert-Ludwigs-University of Freiburg

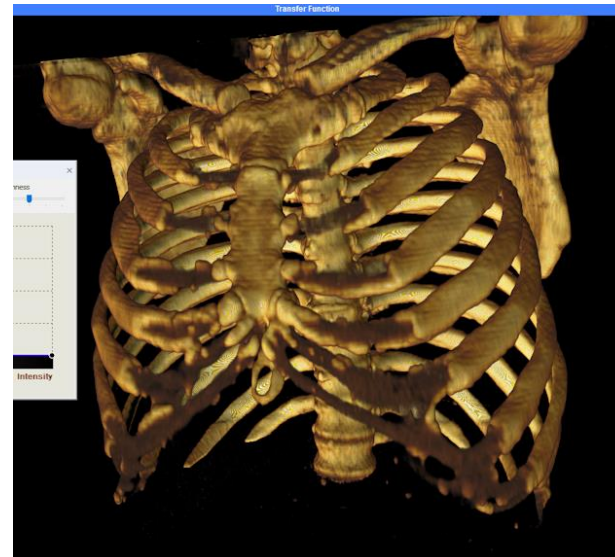
UNI
FREIBURG

Überblick

- Motivation
- Eingabe- und Ausgabedaten
- Methode
- Implementierung
- Ausblick

Motivation

- Medizinische Techniken (CT, MRT...) zeigen ein 3D Volumen als viele 2D Schichtbilder
- Marching Cubes erstellt eine 3D Polygon Oberflächen Repräsentation der Daten mit hohem Detailgrad

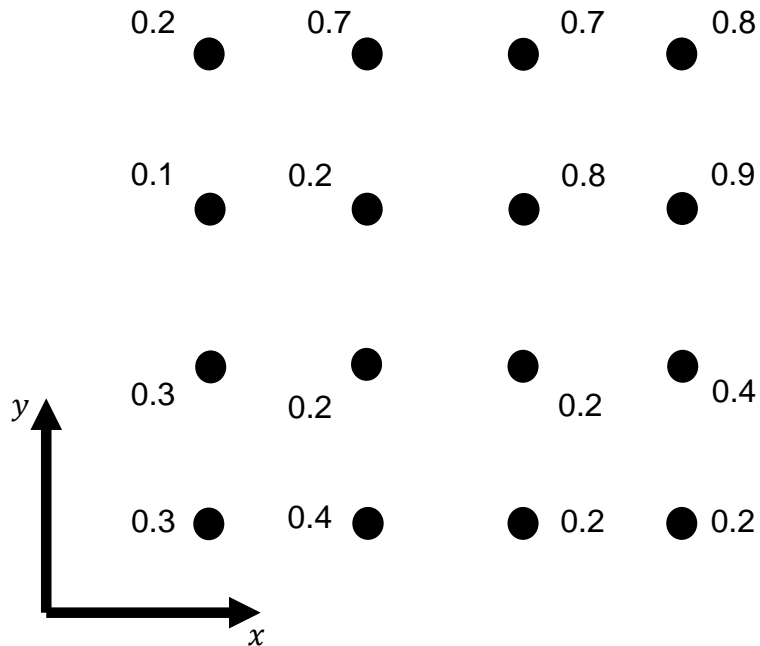


[1]

Eingabe- und Ausgabedaten

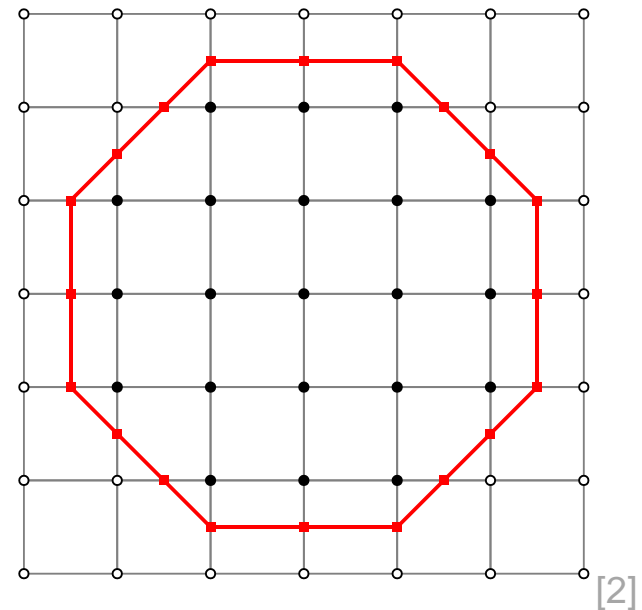
Eingabe

- Dichtewerte $D(x, y)$
- Isowert ρ_{iso}

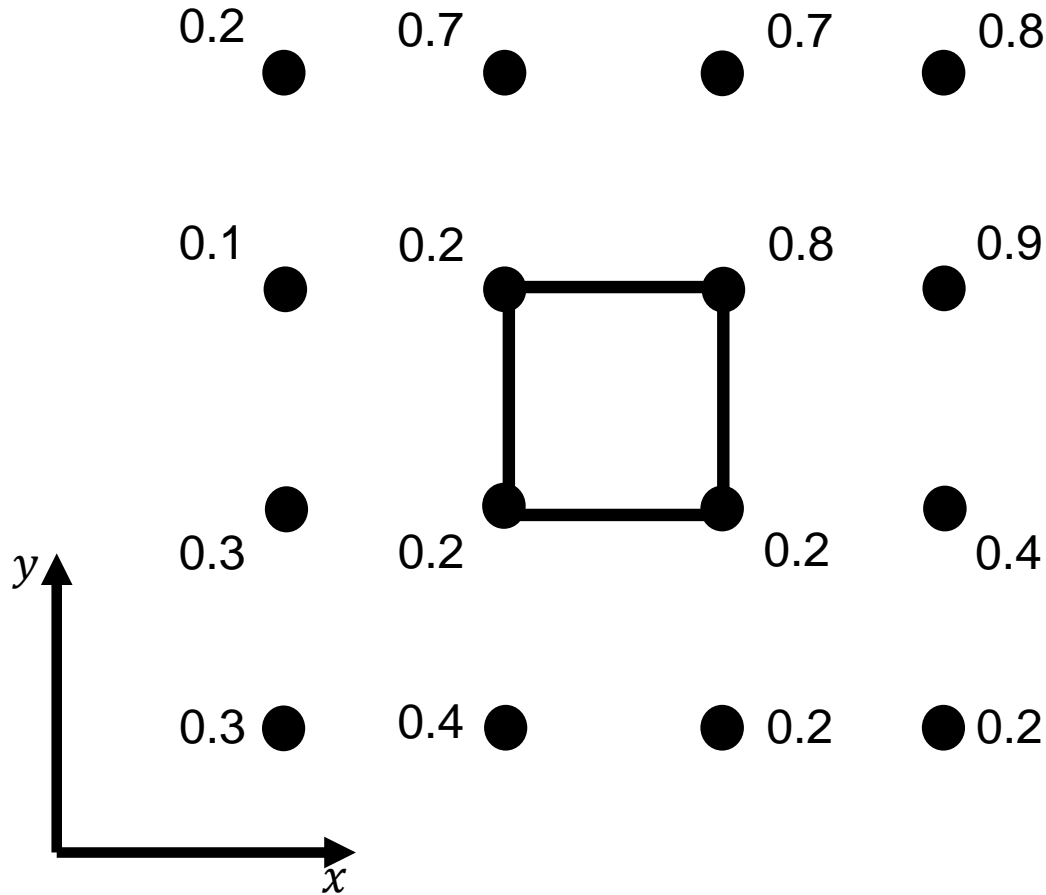


Ausgabe

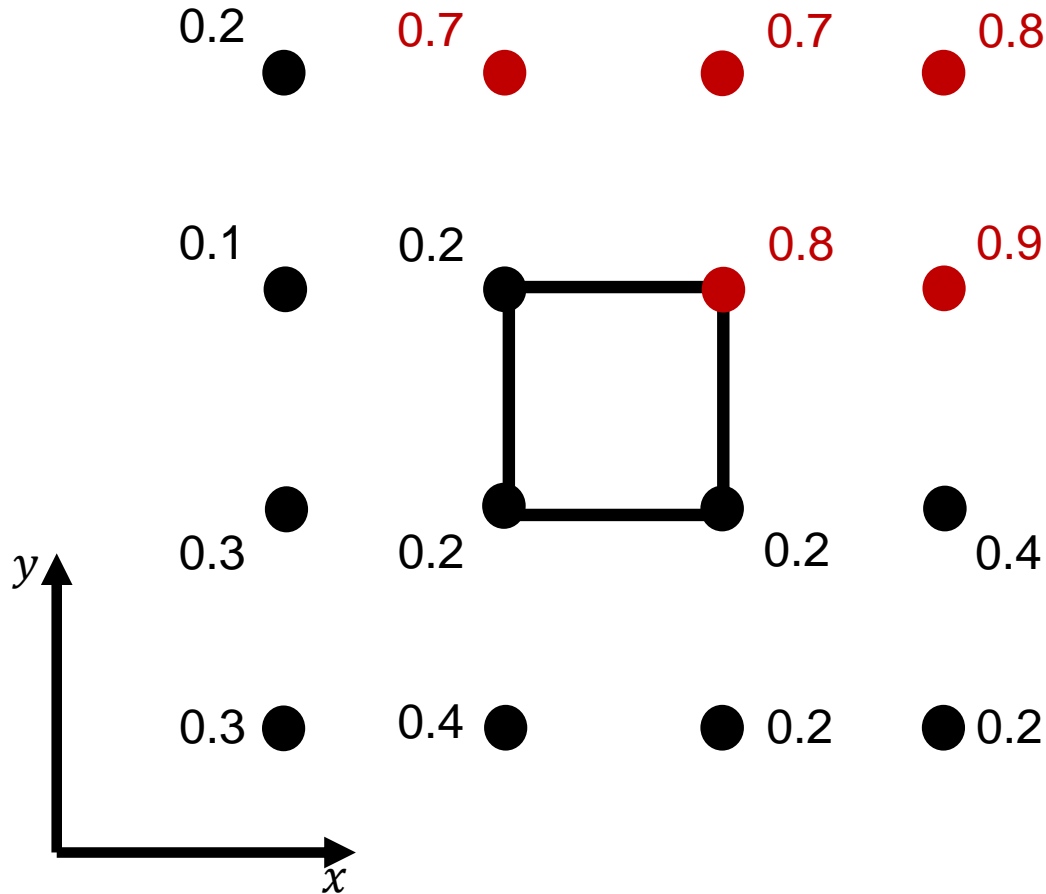
- Isofläche
- Zugehörige Normalen der Eckpunkte



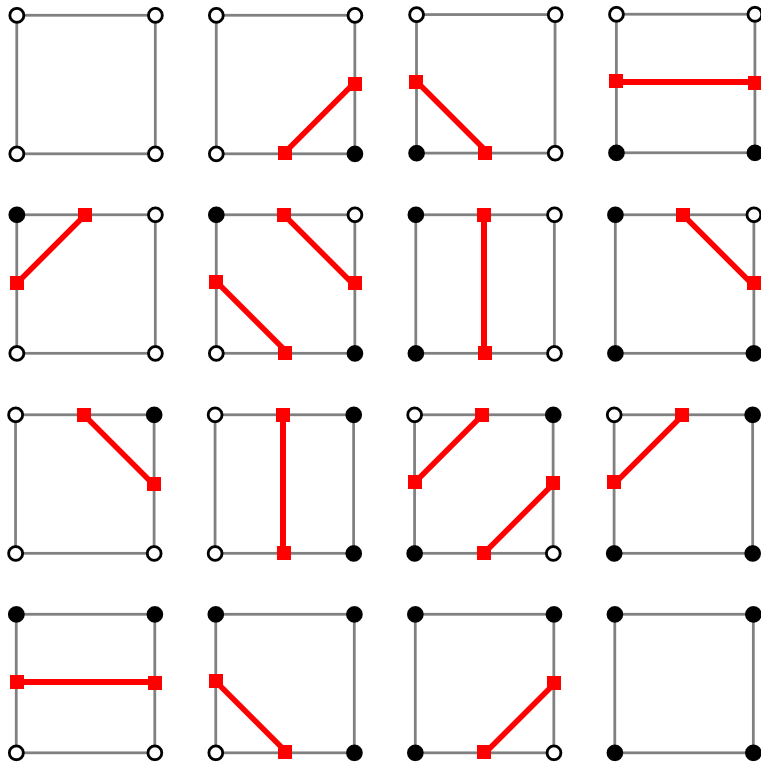
Methode



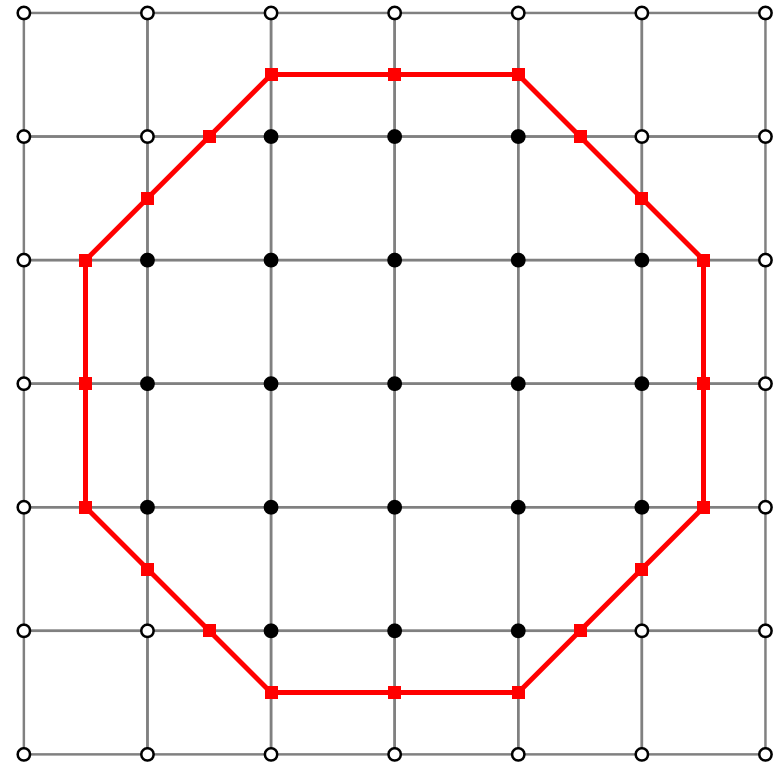
Aktive und Inaktive Ecken



Mögliche Kombinationen

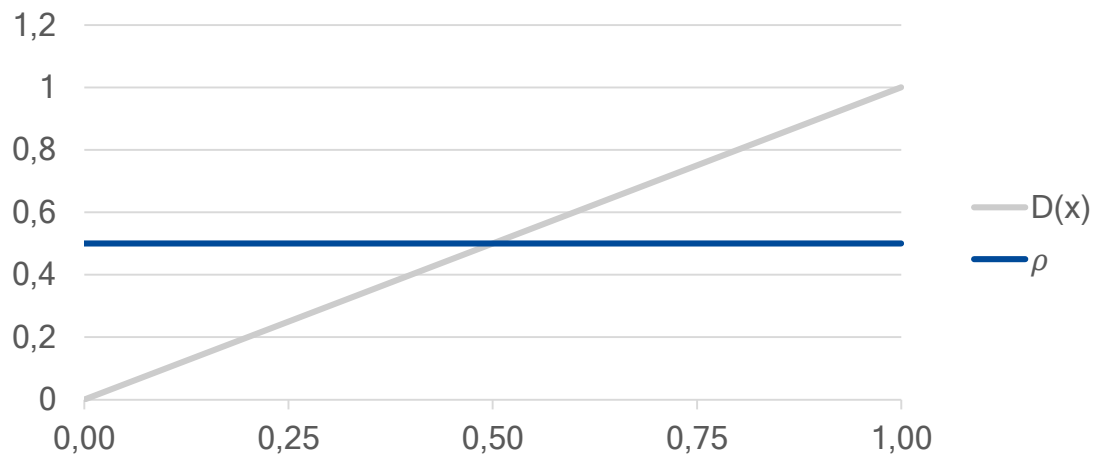
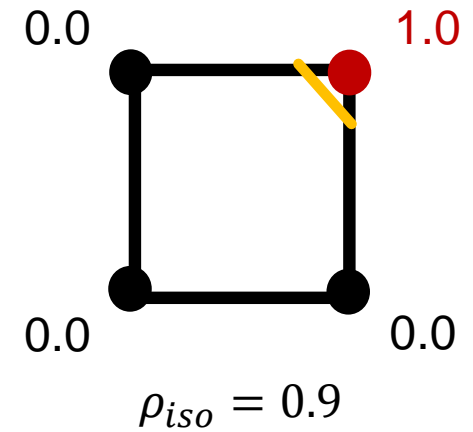
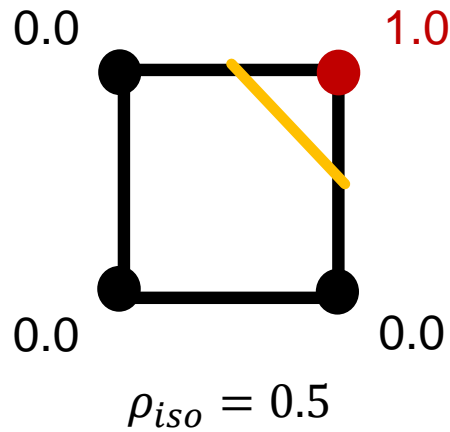
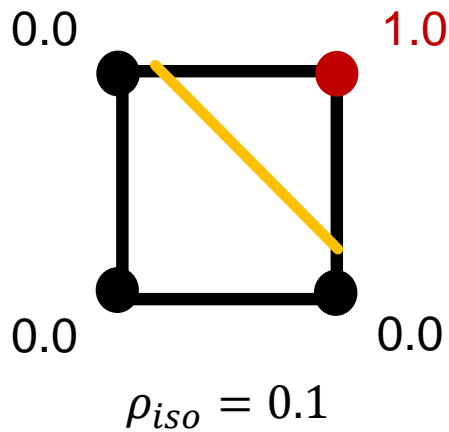


[3]



[2]

Interpolation



Normale

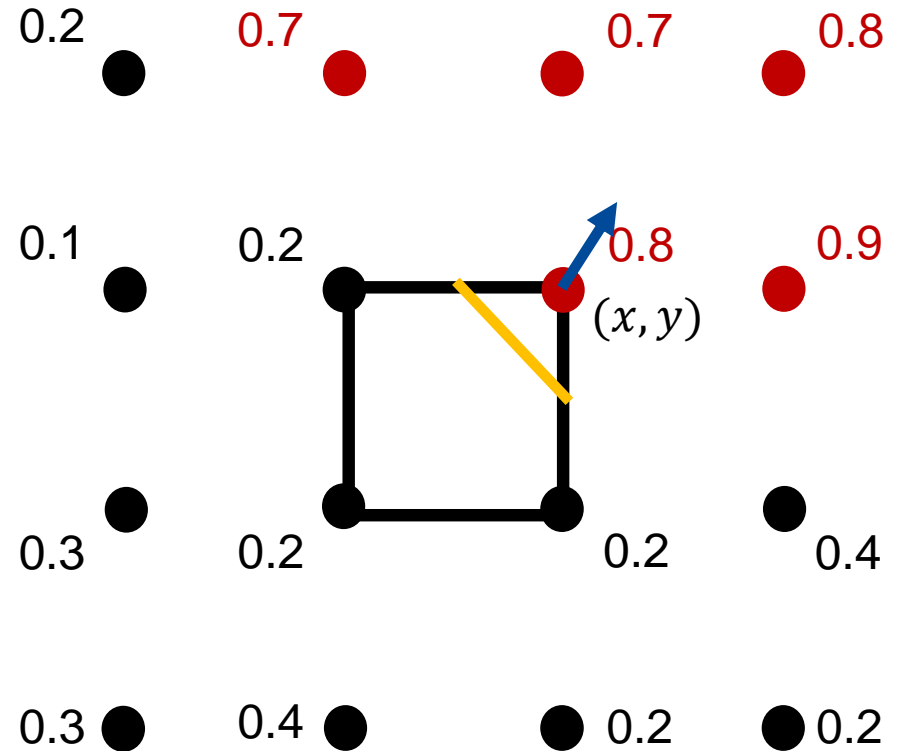
$D(x, y)$: Dichtewert an der Stelle (x, y)

$G(x, y)$: Gradient an der Stelle (x, y)

$$G_x(x, y) = \frac{D(x + 1, y) - D(x - 1, y)}{2\Delta x}$$

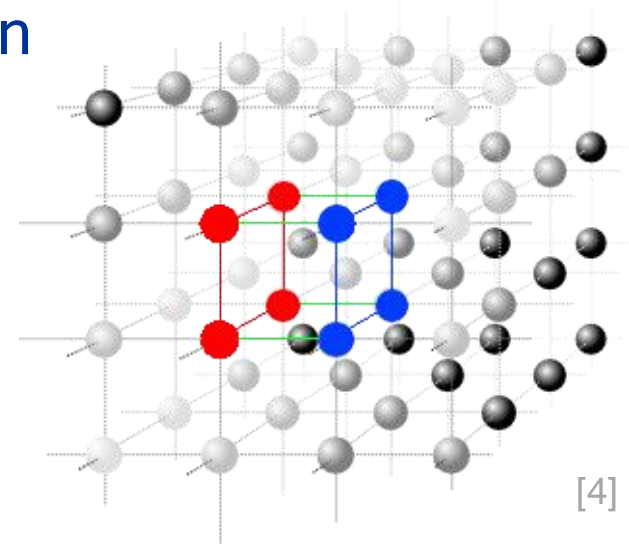
$$G_y(x, y) = \frac{D(x, y + 1) - D(x, y - 1)}{2\Delta y}$$

$$G(x, y) = \begin{pmatrix} G_x \\ G_y \end{pmatrix} = \begin{pmatrix} \frac{0.9 - 0.2}{2 * 1.0} \\ \frac{0.7 - 0.2}{2 * 1.0} \end{pmatrix} = \begin{pmatrix} 0.35 \\ 0.25 \end{pmatrix}$$



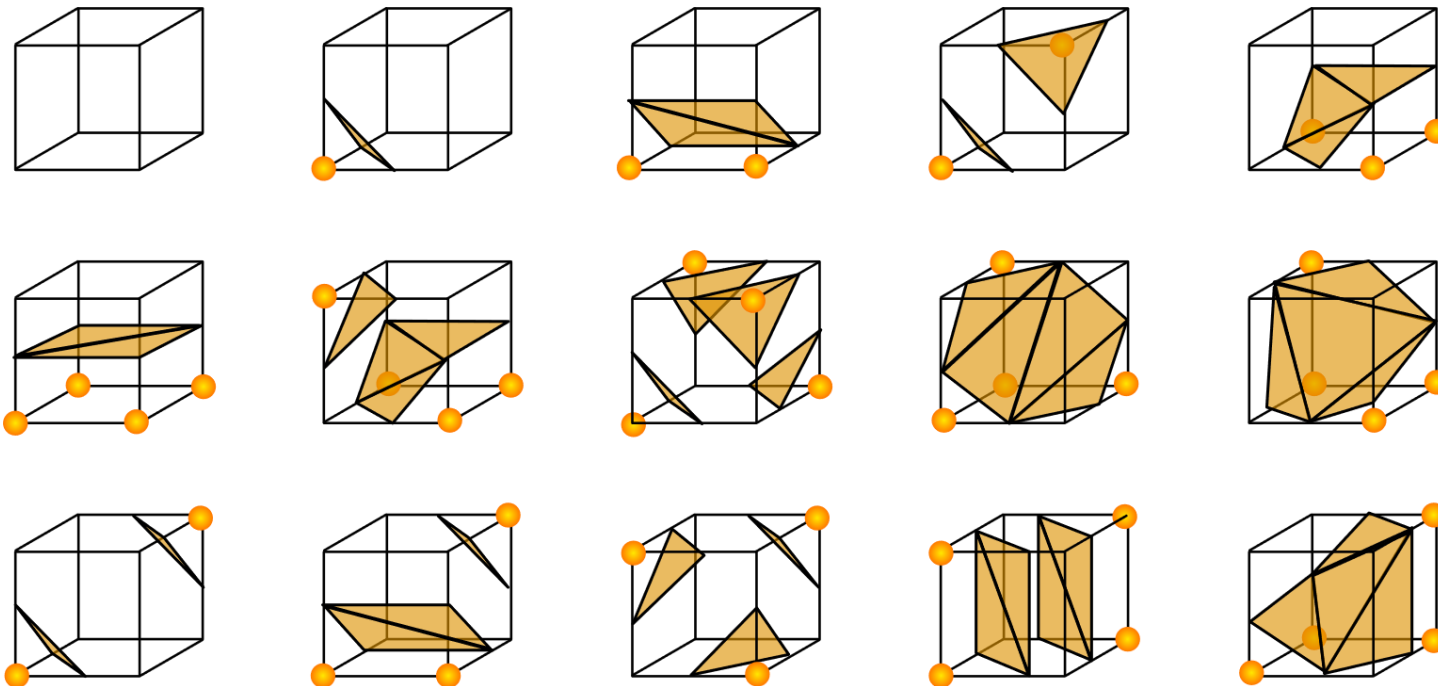
Implementierung

1. Lade 4 aufeinanderfolgende Scheiben
2. Erstelle einen Würfel
3. Ermittle Aktive und Inaktive Ecken
4. Ermittle die Dreiecke im Würfel
5. Interpoliere die Berührungspunkte
6. Berechne Normalen
7. Gebe die Dreiecke und Normalen zurück



Dreieck Lookup Tabelle

- Bildet Indizes auf entsprechende Dreieckskonstellationen ab
- Gibt diese entsprechend (ggf. rotiert) zurück



3D-Normalen

- Normalen entsprechen negativem Dichtegradienten
- Gradienten an den Eckpunkten der Würfel mittels Central-Difference berechnen
- Zwischen Gradienten linear interpolieren

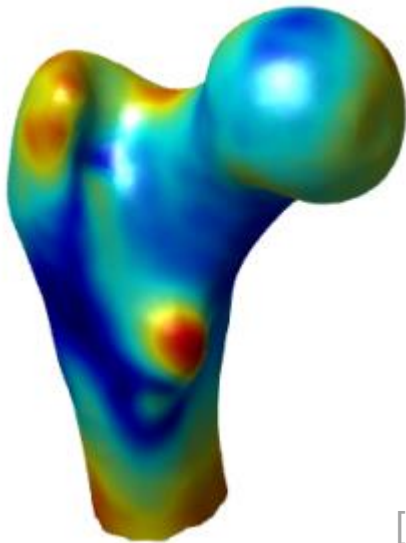
$$G_x(x, y, z) = \frac{D(x + 1, y, z) - D(x - 1, y, z)}{2\Delta x}$$

$$G_y(x, y, z) = \frac{D(x, y + 1, z) - D(x, y - 1, z)}{2\Delta y}$$

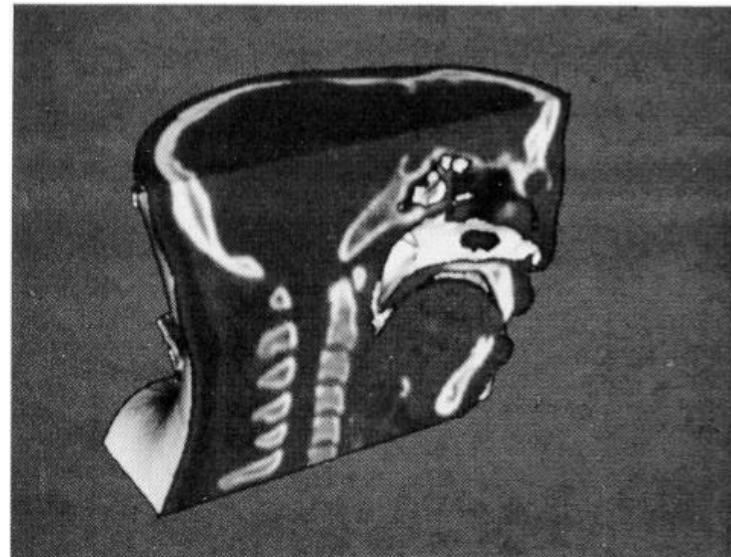
$$G_z(x, y, z) = \frac{D(x, y, z + 1) - D(x, y, z - 1)}{2\Delta z}$$

Ausblick

- Zusätzliche Informationen im Dichtegitter können zusammen mit den Dreiecken ausgegeben werden
- Schnitte durch das Modell können mittels boolescher Operatoren auf die Bits der Würfel umgesetzt werden



[6]



[7]

Zusammenfassung

- Marching Cubes erstellt aus einem Gitter mit Dichtedaten ein trianguliertes 3D-Oberflächenmodell mit hoher Genauigkeit
- Für jeden kleinstmöglichen Würfel im Gitter wird die Isofläche bestimmt
- Normalen werden mittels des Dichtegradienten berechnet
- Der Algorithmus wird zur Darstellung von u.a. MRT und CT Daten verwendet

Bildquellen

- [1] <https://www.santesoft.com/win/sante-dicom-viewer-3d-pro/sante-dicom-viewer-3d-pro.html#prettyPhoto>
- [2] <https://www.boristhebrave.com/2018/04/15/marching-cubes-tutorial/>
- [3] <https://www.boristhebrave.com/2018/04/15/marching-cubes-tutorial/>
- [4] https://de.wikipedia.org/wiki/Marching_Cubes#/media/File:Voxelgitter.png
- [5] https://de.wikipedia.org/wiki/Marching_Cubes#/media/File:MarchingCubes.svg
- [6] http://www.is.uni-due.de/lehrstuhl/mitarbeiter/adrian_morariu/?tx_sevenpack_pi1%5Bshow_uid%5D=1554&cHash=6e3d4a1fad46f898848dfbc5bd9c5c2a
- [7] “Marching Cubes: A High Resolution 3D Surface Construction Algorithm”, William E. Lorensen and Harvey E. Cline, Computer Graphics, Volume 21, Number 4, July 1987, Figure 11 „Sagittal Cut with Texture Mapping.“

Quellen

- “Marching Cubes: A High Resolution 3D Surface Construction Algorithm”, William E. Lorensen and Harvey E. Cline, Computer Graphics, Volume 21, Number 4, July 1987
- “Marching Cubes”, https://de.wikipedia.org/wiki/Marching_Cubes, Letzter Aufruf 04.01.2019 21:36 Uhr
- “Finite difference”, https://en.wikipedia.org/wiki/Finite_difference, Letzter Aufru 04.01.2019 20:41 Uhr
- “Wie lässt sich voxelbasierte 3D-Umgebung polygonisieren?“, Marcus Jahns, „https://www.inf.fu-berlin.de/inst/ag-ki/rojas_home/documents/Betreute_Arbeiten/Bachelor-Jahns.pdf“ Letzter Abruf: 28.12.2018 15:56 Uhr