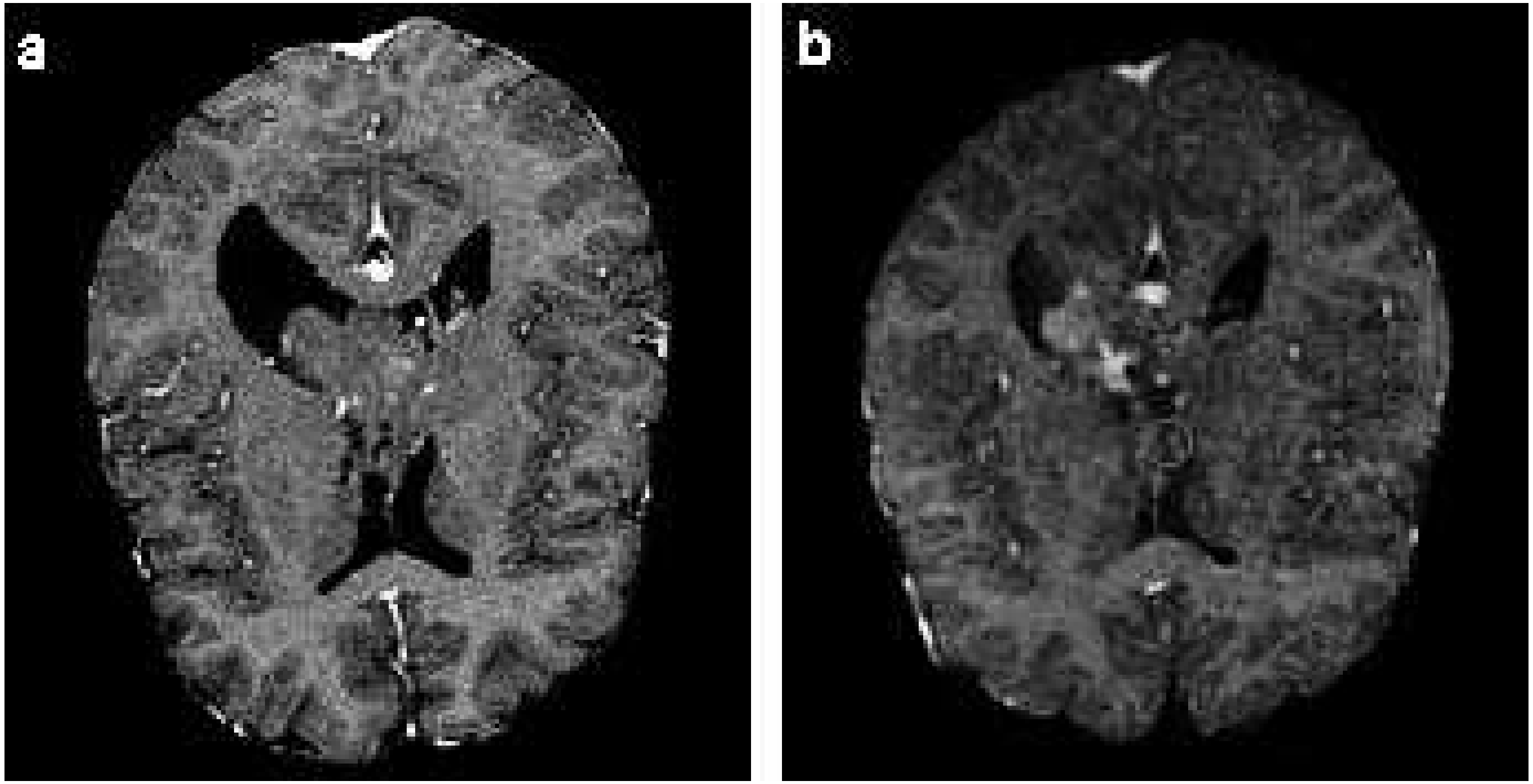
Marching-Cube-Algorithmus

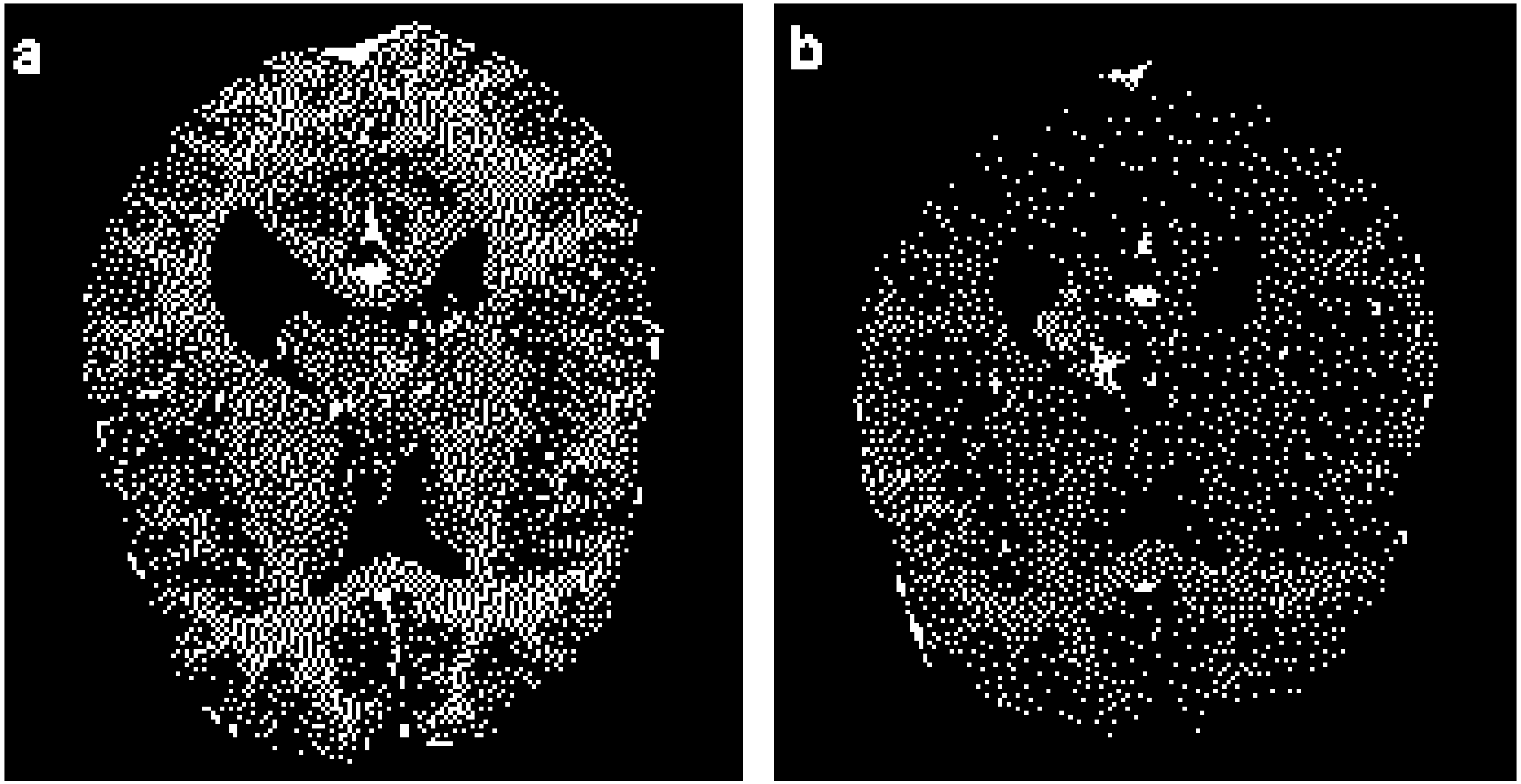
1 Motivation

- vorgeschlagen von E. Lorensen und Harvey E. Cline 1987
- entwickelt für die medizinische Bildverarbeitung
- wird verwendet um z.B. aus CT-Schichtbildern hochauflösende, dreidimensionale Oberflächenmodelle zu erstellen

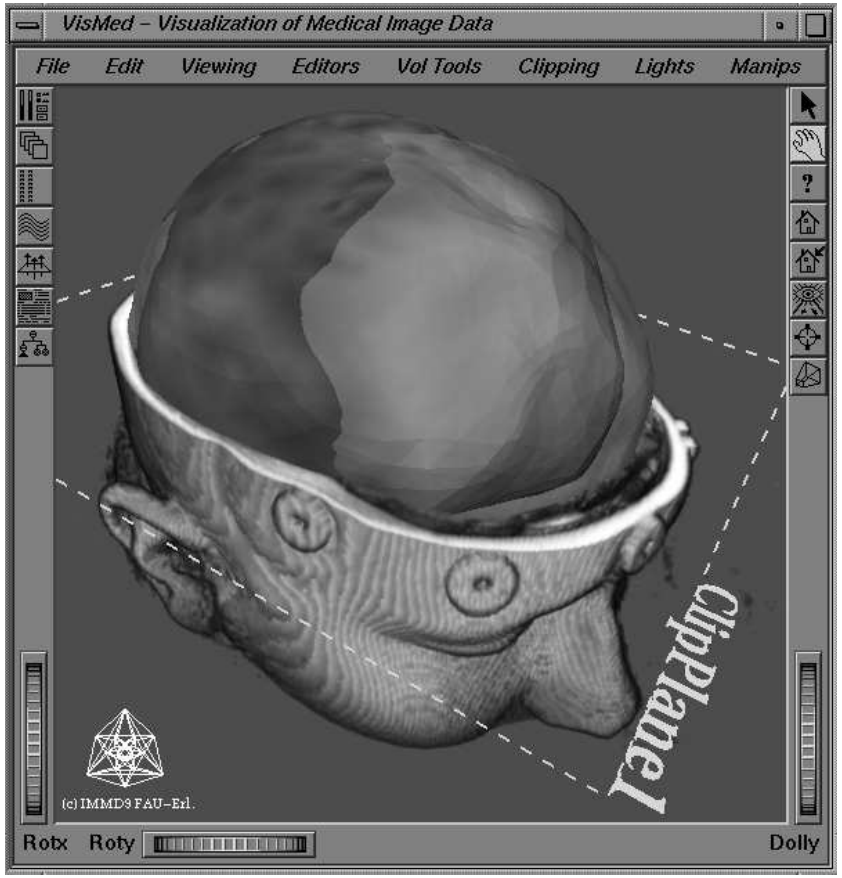
Beispiel: Schichtbilder...



...werden gerastert...



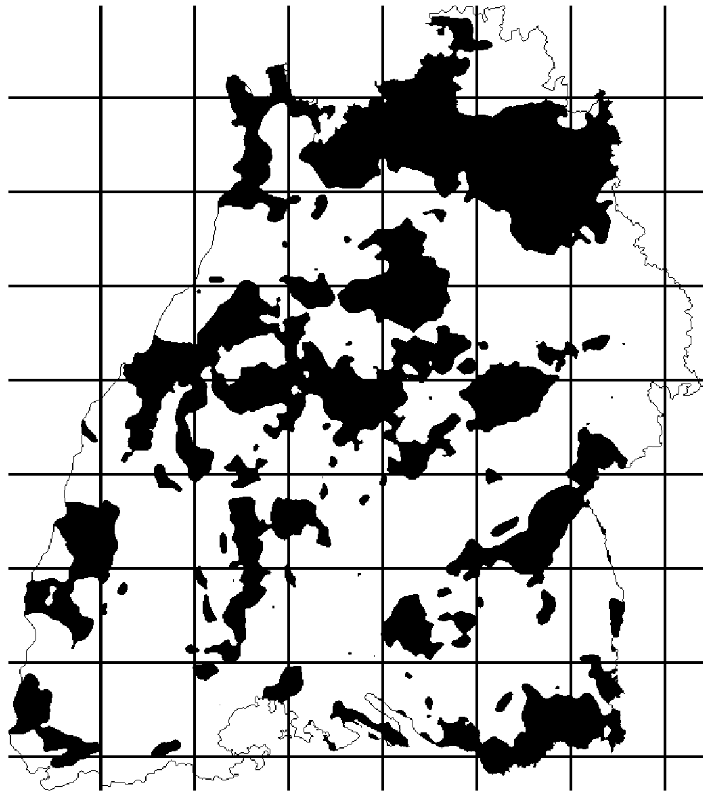
...um daraus entsprechende Oberflächenmodelle zu berechnen!



2 Der Marching-Square-Algorithmus

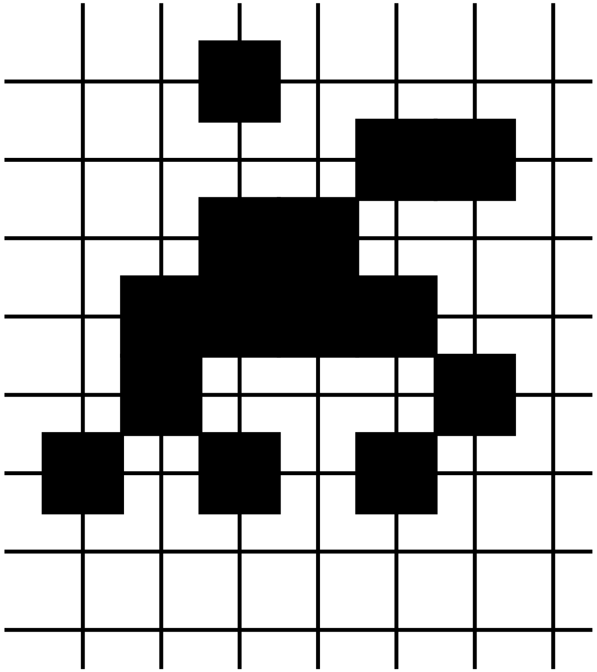
- im zweidimensionalen Raum entsteht anstelle einer Oberfläche eine Kontur
- in unserem Fall werden keine Schichtbilder mit unterschiedlichen Grauwerten, sondern ausschließlich Binärbilder (schwarz/weiß Bilder) betrachtet
- beim Marching-Square-Algorithmus wird zunächst über ein kontinuierliches Bild ein Gitternetz gelegt, welches das Bild in Quadrate unterteilt

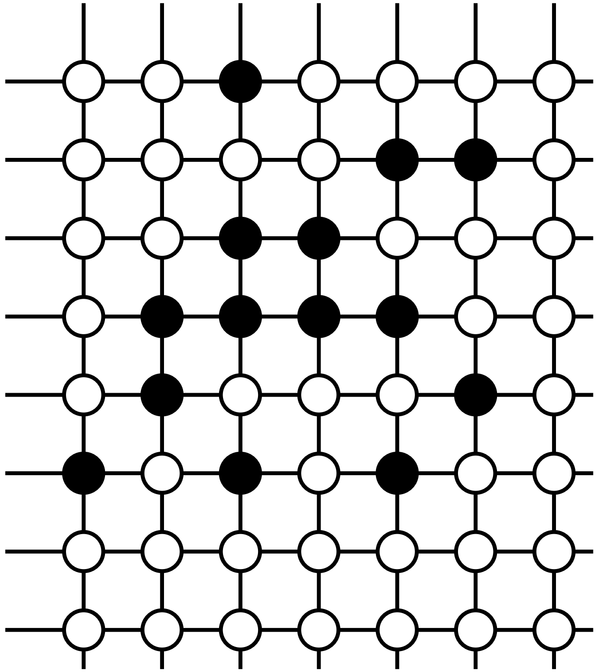
Kontinuierliches Bild und Gitternetzeinteilung



- anschließend wird überprüft, ob an den Kreuzungspunkten des Gitternetzes ein Punkt liegt oder nicht
- ist ein Punkt vorhanden, so wird die zu diesem Kreuzungspunkt gehörende Fläche schwarz gefärbt, im anderen Fall bleibt sie weiß
- **Achtung:** die Fläche wird unabhängig vom Flächenanteil des kontinuierlichen Bildes eingefärbt - nur der direkt auf dem Kreuzungspunkt liegende Punkt bestimmt die Farbe
- **Problem:** bei sehr kleinkörnigen Bildern kann die Form des Bildes ein vollständig verändertes Aussehen erhalten
- **Verbesserung:** das Gitternetz - wenn möglich - solange verfeinern, bis die gewünschte Form erreicht ist...

Gerasterte Bilder

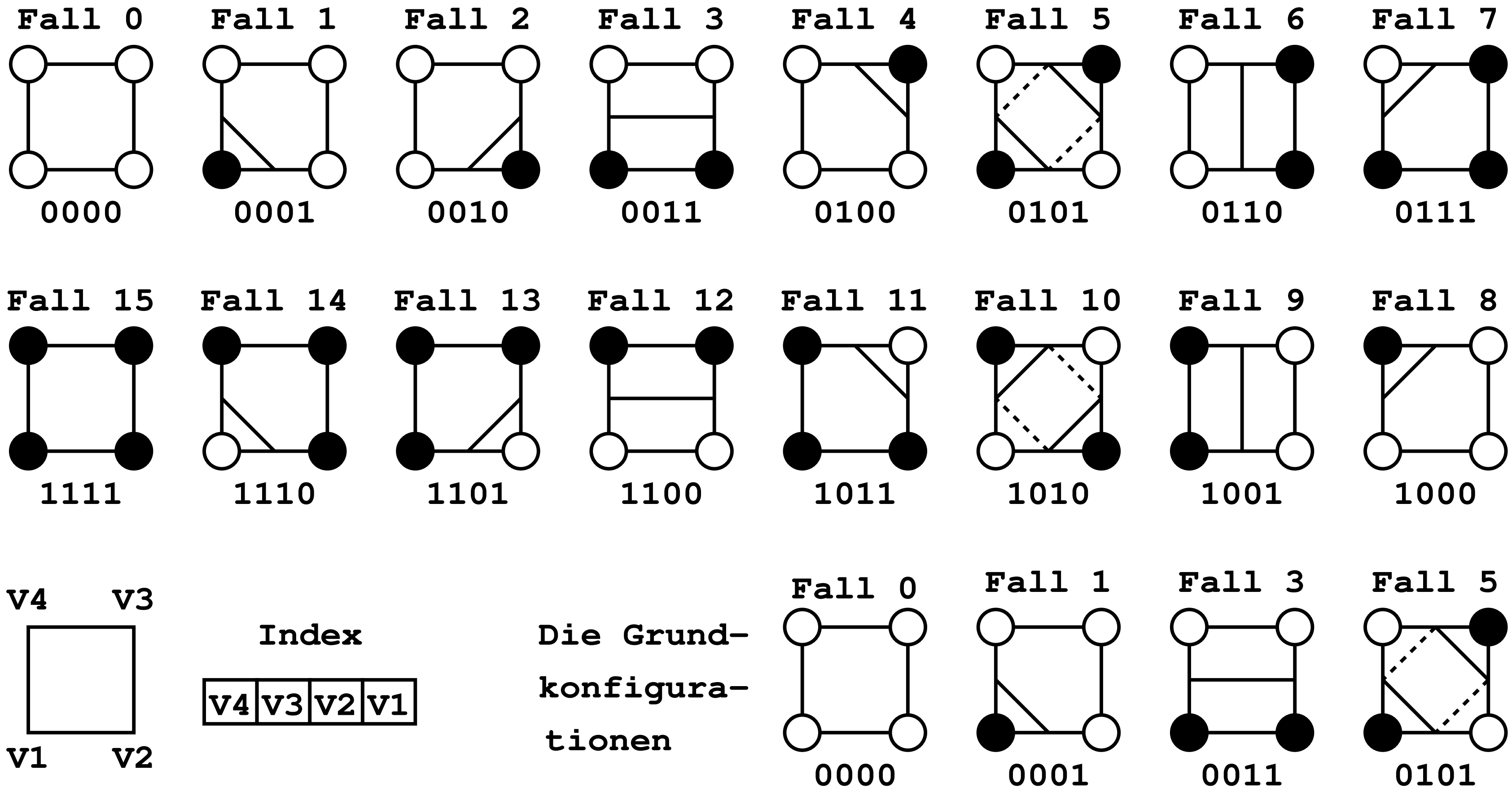




- im nächsten Schritt wird die Kontur um die schwarzen Flächen berechnet
- beim Marching-Square-Algorithmus existieren $2^4 = 16$ Konfigurationen, wie eine Kontur ein Quadrat des Gitternetzes schneiden kann
- diese 16 Konfigurationen lassen sich durch
  - Rotation
  - Spiegelung
  - Inversion

  auf 4 Grundkonfigurationen zurückführen

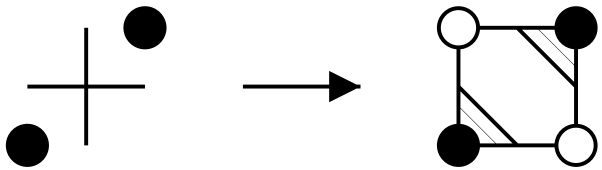
Die 16 Konfigurationen des Marching-Square-Algorithmus



- in Fall 5 und 10 existieren 2 unterschiedliche Schnittmöglichkeiten
- anhand verschiedener Nachbarschaftsbeziehungen (4er, 6er, 8er) wird in diesen Fällen entschieden, wie die Kontur die Quadrate des Gitternetzes schneidet
- mit Hilfe von Fall 5 wird im Folgenden gezeigt, wie die Schnitte unter den oben genannten Nachbarschaftsbeziehungen aussehen
- anschließend sind die daraus resultierenden Konturzüge dargestellt

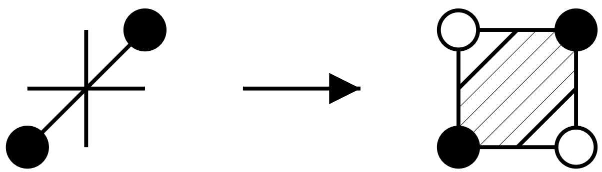
Schnittmöglichkeiten

4er-Nachbarschaft



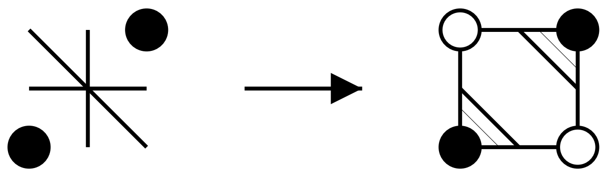
6er-Nachbarschaft

Fall 1

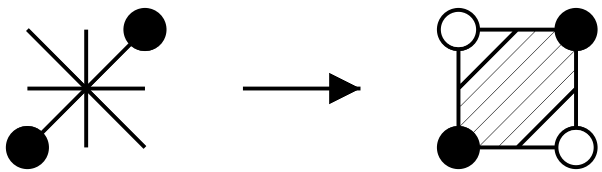


6er-Nachbarschaft

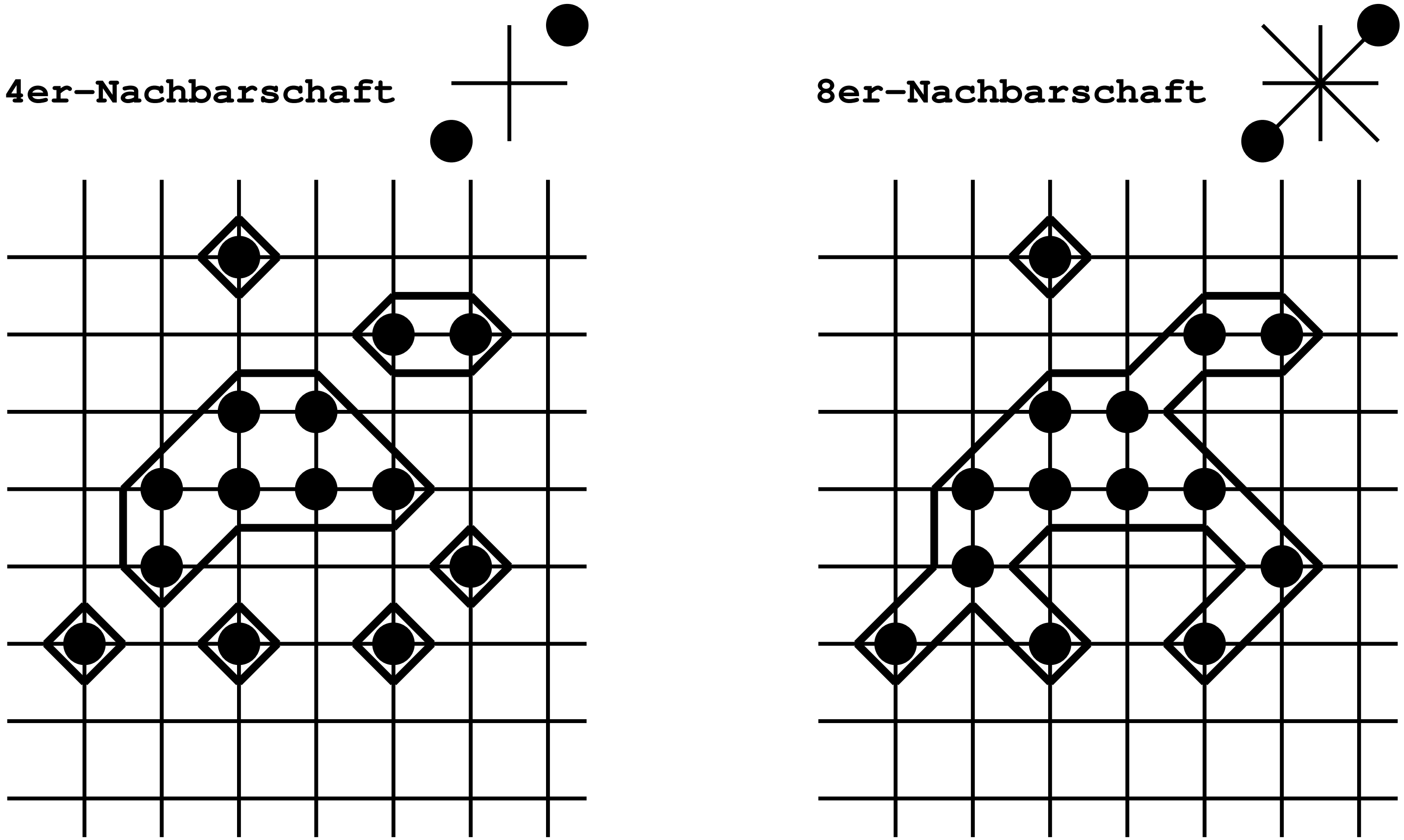
Fall 2



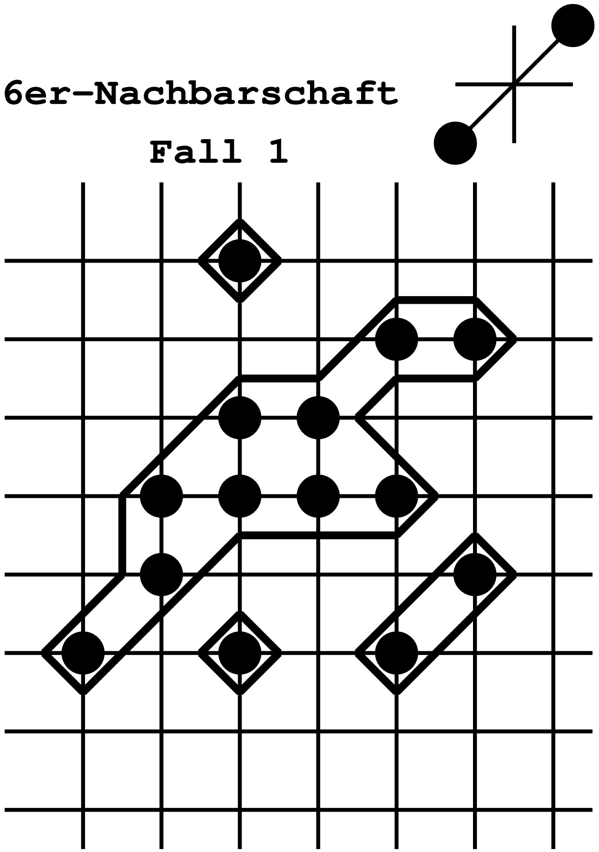
8er-Nachbarschaft

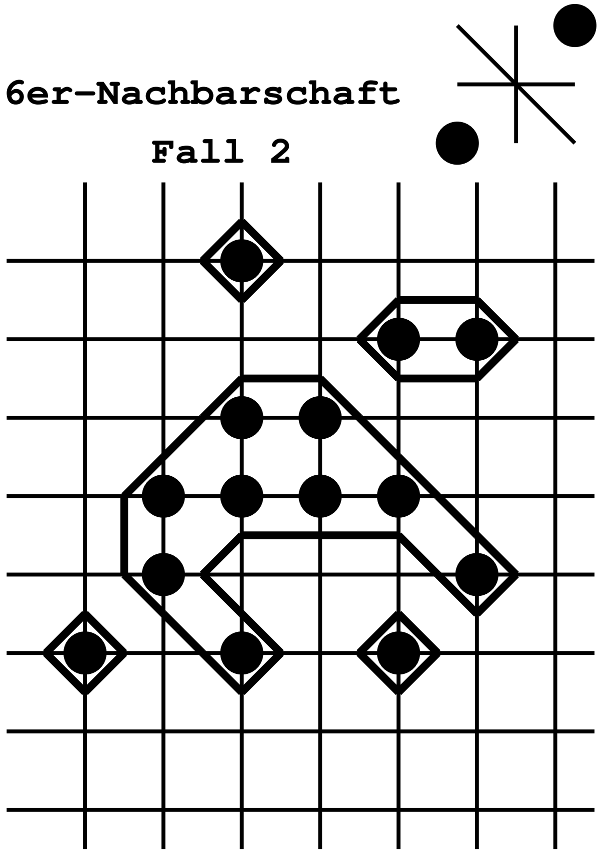


Konturzüge bei 4er- und 8er-Nachbarschaft



Konturzüge bei 6er-Nachbarschaften





Zusammenfassung Marching-Square-Algorithmus

- Gitternetz erstellen
- Bild rastern bzw. Eckpunkte der Gitternetzquadrate berechnen
- für jedes Gitternetzquadrat ergibt sich anhand der nicht gesetzten (0) bzw. gesetzten (1) Eckpunkte ein 4-bit Index
- durch Vergleich der Indizes mit den Konfigurationen wird festgelegt, wie die Kontur das jeweilige Quadrat schneidet
- dementsprechend kann die Kontur gezeichnet werden

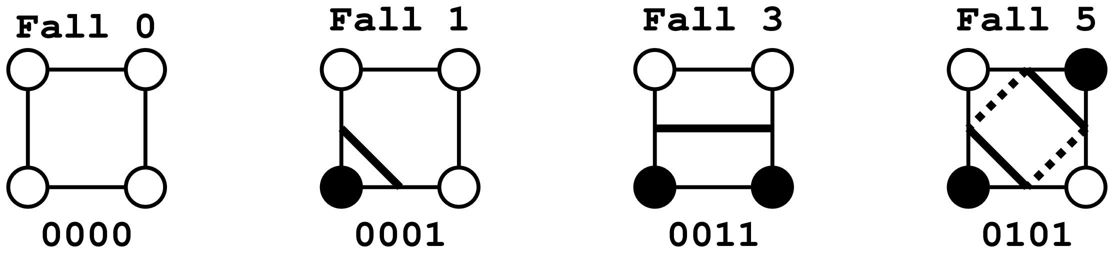
3 Marching-Square-Algorithmus - Maßzahlen

Randlänge

- sei M die Menge aller möglichen Konfigurationen
- sei $h_i, i = 0, \ldots, 15$ die Häufigkeit einer bestimmten Grundkonfiguration i bzgl. eines Bildes Ξ
- sei l_i die Randlänge einer bestimmten Konfiguration i
- dann ist Folgendes ein Schätzer für die Randlänge $L(\Xi)$:

$$\hat{L}(\Xi) = \sum_{i=0}^{15} h_i \cdot l_i$$

Randlängen l der Grundkonfigurationen (bzgl. Seitenlänge a)



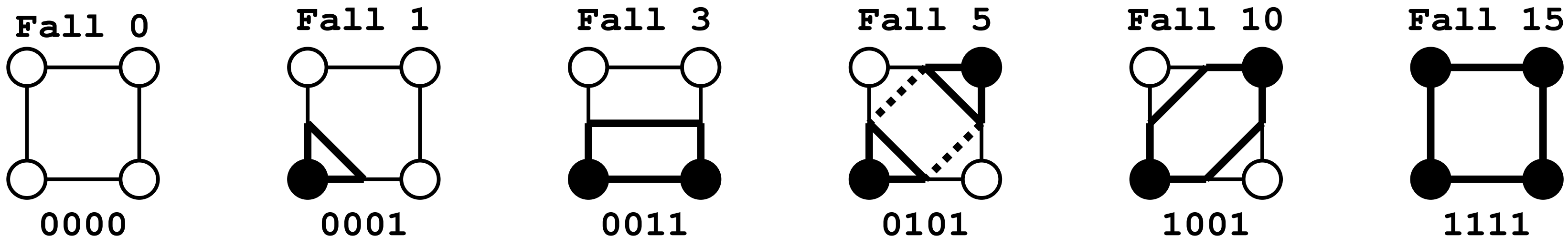
0	$\frac{1}{2}\sqrt{2} \cdot a$	a	$\sqrt{2} \cdot a$

Fläche

- sei M die Menge aller möglichen Konfigurationen
- sei $h_i, i = 0, \ldots, 15$ die Häufigkeit einer bestimmten Konfiguration i bzgl. eines Bildes Ξ
- sei a_i die Fläche einer bestimmten Konfiguration i
- dann ist Folgendes ein Schätzer für die Fläche $A(\Xi)$:

$$\hat{A}(\Xi) = \sum_{i=0}^{15} h_i \cdot a_i$$

Flächen a einiger Konfigurationen (bzgl. Seitenlänge a)



0	$\frac{1}{8} \cdot a^2$	$\frac{1}{2} \cdot a^2$	$\frac{1}{4} \cdot a^2$	$\frac{3}{4} \cdot a^2$	a^2

$$bzw. \quad \frac{3}{4} \cdot a^2$$

Zusammenfassung

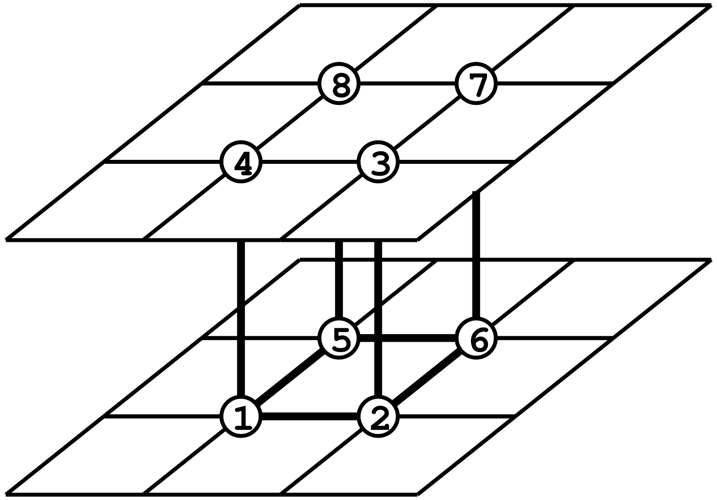
- durch Ermitteln der Konfigurationshäufigkeiten h_i eines Bildes Ξ und **einmaliger** Berechnung (Tabelle) der
  - Randlängen l_i sowie der
  - Flächen a_i

  ist es möglich, mit Hilfe von 2 einfachen Formeln die Randlänge L und die Fläche A des entsprechenden Bildes Ξ zu schätzen:
  - $\hat{L}(\Xi) =< h_i, l_i >$ bzw.
  - $\hat{A}(\Xi) =< h_i, a_i >$

4 Der Marching-Cube-Algorithmus

- ausgehend von 3-dimensionalen Bildern werden beim Marching-Cube-Algorithmus Oberflächen berechnet
- im Gegensatz zum Marching-Square-Algorithmus wird hier das 3-dimensionale Bild durch eine Gitterstruktur in Quader eingeteilt
- dabei werden aus 2 übereinanderliegenden Schichten jeweils 4 übereinanderliegende Pixel ausgewählt und hieraus ein Quader gebildet

Quaderbildung beim Marching-Cube-Algorithmus

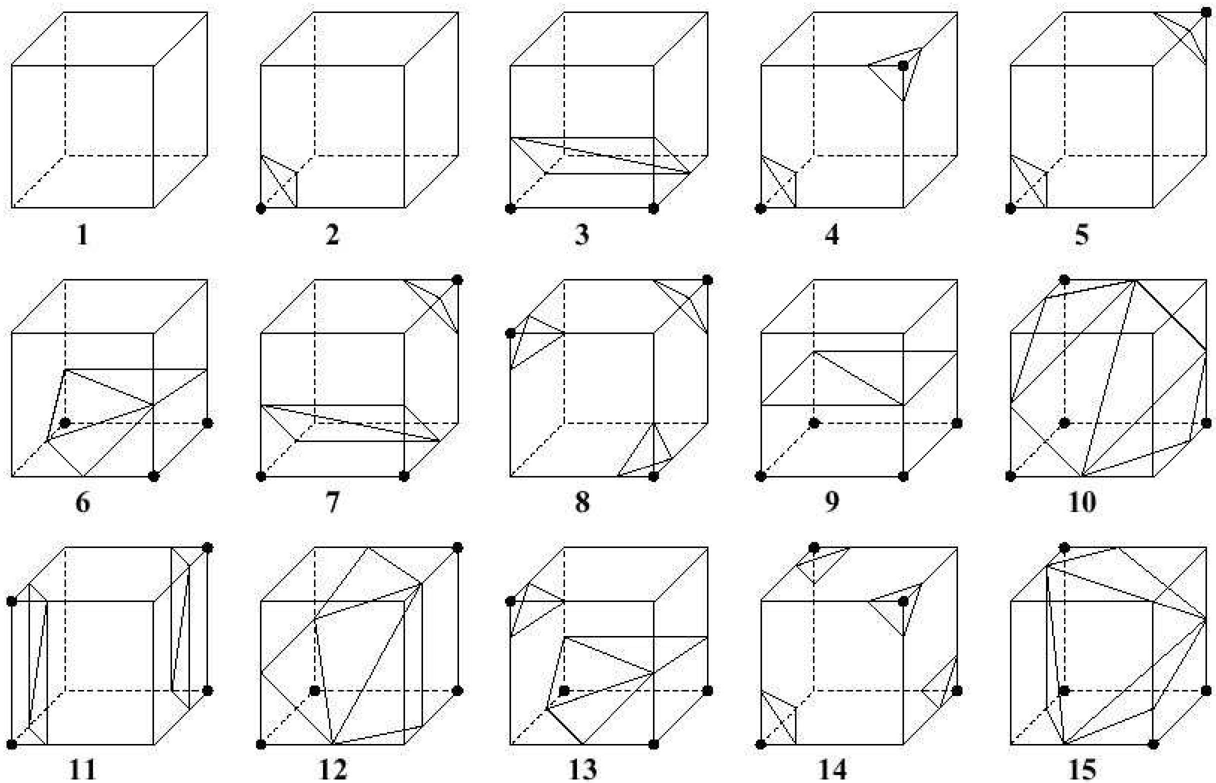


Index

V8	V7	V6	V5	V4	V3	V2	V1

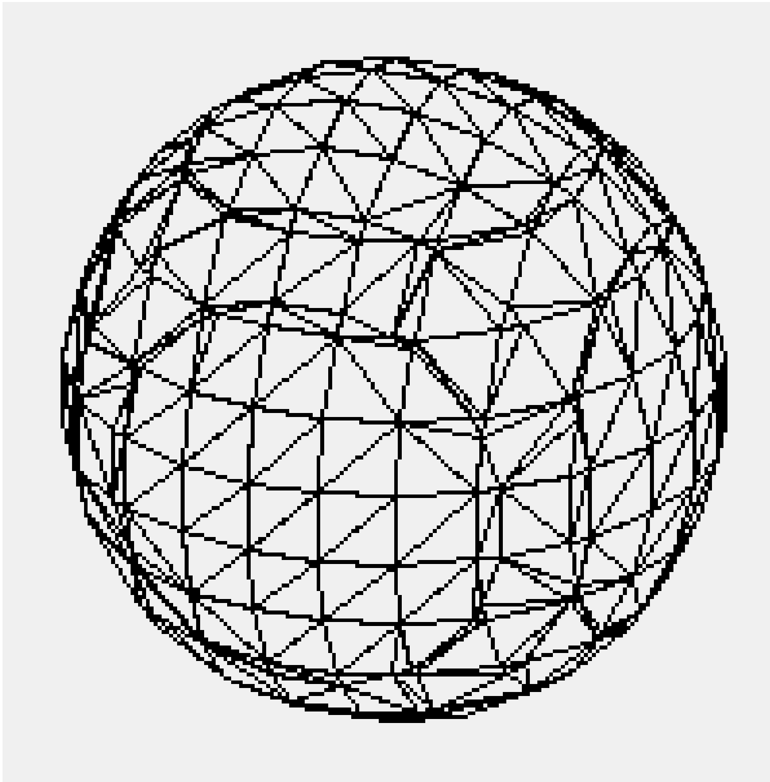
- analog zum Marching-Square-Algorithmus wird anschließend überprüft, ob an den Kreuzungspunkten der Gitterstruktur ein Punkt liegt oder nicht
- dementsprechend werden die Quader eingefärbt
- durch den 8-bit Index ergeben sich beim Marching-Cube-Algorithmus 256 Konfigurationen
- diese können wiederum durch die Symmetrieeigenschaften Rotation, Spiegelung und Inversion auf 15 Grundkonfigurationen reduziert werden

Die von Lorensen und Cline vorgeschlagenen 15 Grundkonfigurationen



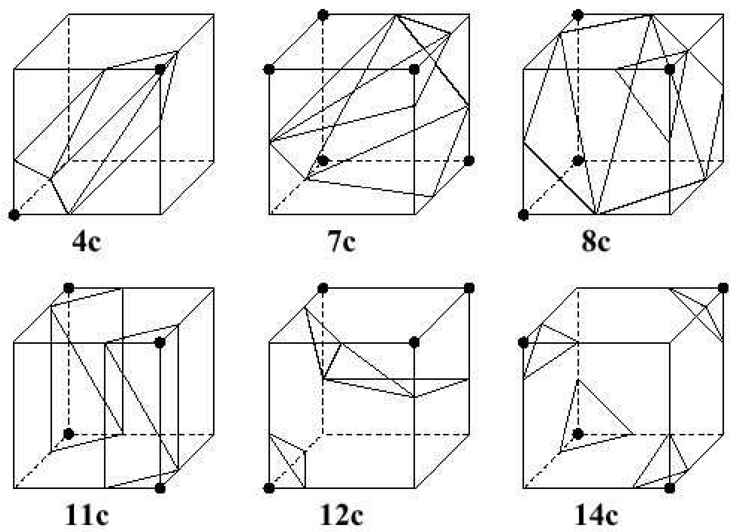
- durch Durchwandern der einzelnen Quader eines 3D-Bildes (marching) und Vergleichen der Indizes mit den Konfigurationen kann ein Polygonnetz aufgespannt werden
- die so entstandene Oberfläche ist jedoch sehr kantig
- sie kann durch lineare Interpolation geglättet werden, d.h. die Oberfläche schneidet die Quaderkanten nicht mehr genau in der Mitte, sondern etwas weiter zu den Ecken hin verschoben, wie bereits im vorhergehenden Bild gezeigt

Oberflächenberechnung einer Kugel nach Lorensen und Cline



- ebenso wie beim Marching-Square-Algorithmus können Konfigurationen durch unterschiedliche Nachbarschaftsbeziehungen (6er, 14er, 26er) mehrdeutig sein
- zusätzlich können Löcher entstehen, die beim Marching-Square-Algorithmus durch geschickte Nachbarschaftswahl aufgelöst werden können, hier jedoch nicht mehr auflösbar sind
- von W. J. Schroeder et al. (1992) wurden deshalb zusätzliche Konfigurationen vorgeschlagen, um dieses Problem zu lösen

Zusätzliche vorgeschlagene Konfigurationen von Schroeder et al.



Zusammenfassung Marching-Cube-Algorithmus

- Gitterstruktur erstellen
- Quadereckpunkte einfärben
- hieraus entsteht für jeden Quader ein 8-bit Index
- durch Vergleich der Indizes mit den Konfigurationen wird festgelegt, wie die Oberfläche den jeweiligen Quader schneidet
- dementsprechend kann die Oberfläche berechnet werden

5 Marching-Cube-Algorithmus - Maßzahlen

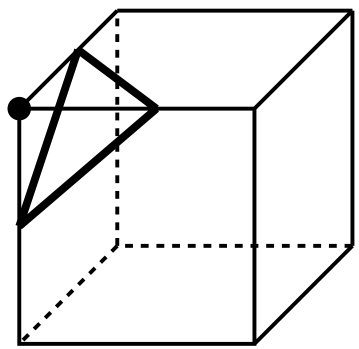
Oberfläche

- sei M die Menge aller möglichen Konfigurationen
- sei $h_i, i = 0, \ldots, 255$ die Häufigkeit einer bestimmten Konfiguration i bzgl. eines Bildes Ξ
- sei a_i die Oberfläche einer bestimmten Konfiguration i
- dann ist Folgendes ein Schätzer für die Oberfläche $A(\Xi)$:

$$\hat{A}(\Xi) = \sum_{i=0}^{255} h_i \cdot a_i$$

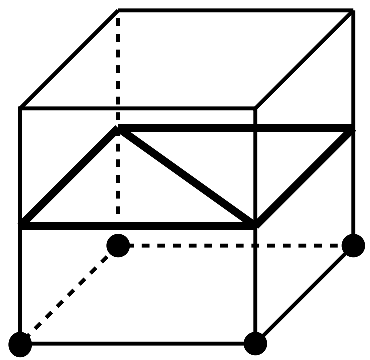
Oberflächen a einiger Konfigurationen (bzgl. Seitenlänge a und 6er-Nachbarschaft)

Fall 2



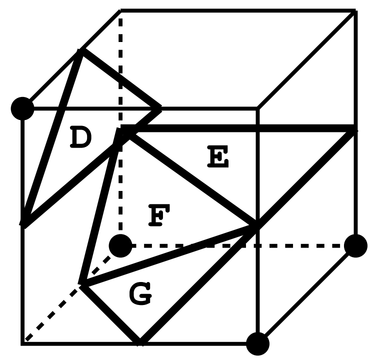
$\sqrt{\frac{3}{8}} \cdot a^2$

Fall 9



a^2

Fall 13



$D + E + F + G$

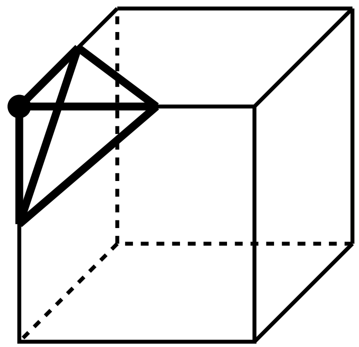
Volumen

- sei M die Menge aller möglichen Konfigurationen
- sei $h_i, i = 0, \ldots, 255$ die Häufigkeit einer bestimmten Konfiguration i bzgl. eines Bildes Ξ
- sei v_i das Volumen einer bestimmten Konfiguration i
- dann ist Folgendes ein Schätzer für das Volumen $V(\Xi)$:

$$\hat{V}(\Xi) = \sum_{i=0}^{255} h_i \cdot v_i$$

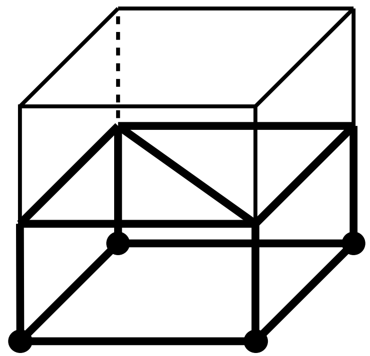
Volumina v einiger Konfigurationen (bzgl. Seitenlänge a und 6er-Nachbarschaft)

Fall 2



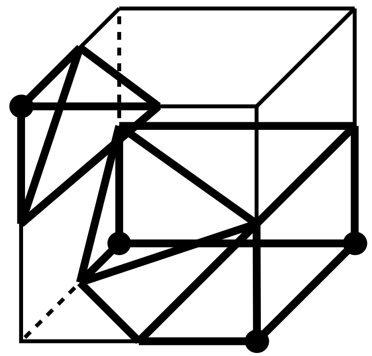
$\frac{a^3}{48}$

Fall 9



$\frac{a^3}{2}$

Fall 13



...

Zusammenfassung

- durch Ermitteln der Konfigurationshäufigkeiten h_i eines Bildes Ξ und **einmaliger** Berechnung (Tabelle) der
  - Oberflächen a_i sowie der
  - Volumina v_i

  ist es möglich, mit Hilfe von 2 einfachen Formeln die Oberfläche A und das Volumen V des entsprechenden Bildes Ξ zu schätzen:
  - $\hat{A}(\Xi) =< h_i, a_i >$ bzw.
  - $\hat{V}(\Xi) =< h_i, v_i >$

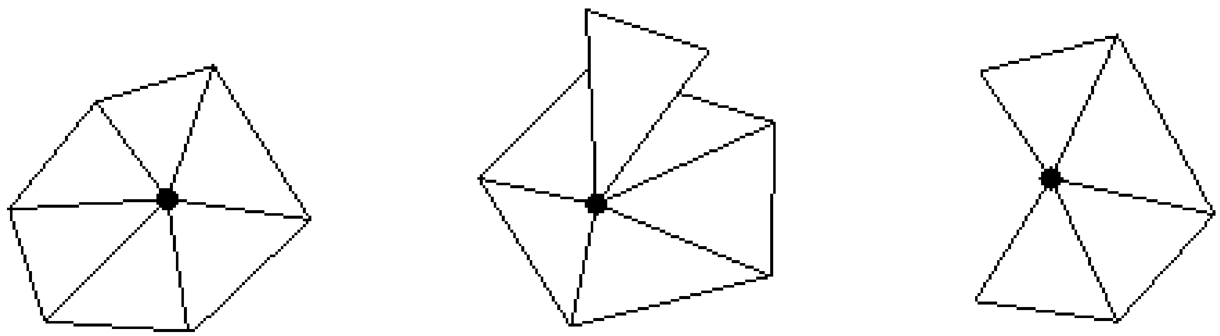
6 Marching-Cube-Algorithmus - Komplexität

Komplexität

- je nach Schnitt werden beim Marching-Cube-Algorithmus 1 bis 4 Dreiecke gebildet, d.h. bei hoher Auflösung ist die Anzahl der Dreiecke, die berechnet werden, beträchtlich
- z.B. können bei Bildern mit 100 Schichten zu 512 x 512 Pixel schon über eine Million Dreiecke entstehen

- für den dreidimensionalen Fall wurde von W. J. Schroeder et al. 1992 ein Approximationsverfahren entwickelt, um die Anzahl der Dreiecke zu reduzieren
- bei dieser Methode werden alle Knotenpunkte des Oberflächengitters anhand ihrer Umgebung klassifiziert
- dann wird je nach Typ und bestimmten Genauigkeitskriterien (Toleranzen) untersucht, ob ein Punkt entfernt werden kann
- anschließend wird die entstandene Lücke durch Triangulierung geschlossen
- dieser Vorgang kann mehrmals wiederholt werden...

Klassifizierung von Punkten

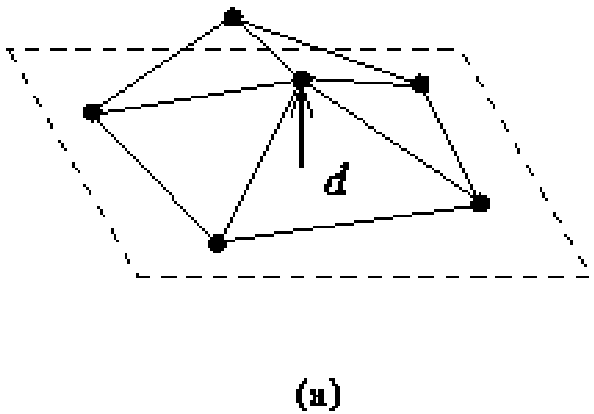


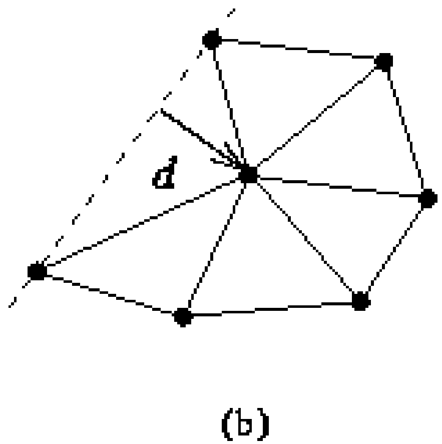
(a) (b) (c)

einfacher Punkt | komplexer Punkt | Kanten-Punkt

- liegt ein *einfacher Punkt* vor, so wird als Kriterium für das Entfernen aus dem Gitter der Abstand zu einer Durchschnittsebene berechnet
- *komplexe Punkte* treten nur an Grenzflächen auf - die hier nicht betrachtet werden - und werden nie gelöscht
- für *Kantenpunkte* wird der Abstand des Punktes zur Gitterkante berechnet. Liegt der Abstand unter einer vorgegebenen Toleranzgrenze, wird dieser Punkt gelöscht und dafür eine neue Gitterkante eingefügt. Anschließend muß das planare Polygon neu trianguliert werden

Entfernen von Punkten



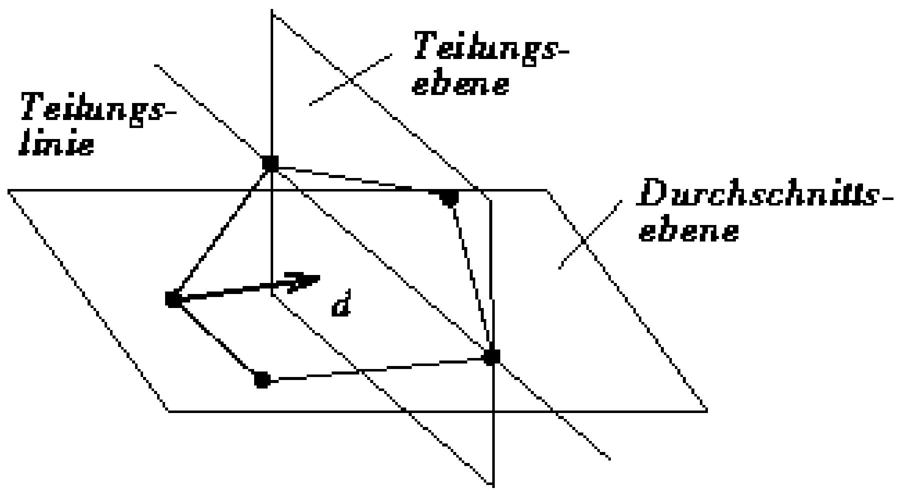


Abstand zu einer Durchschnittsebene

Abstand zu einer Gitterkante

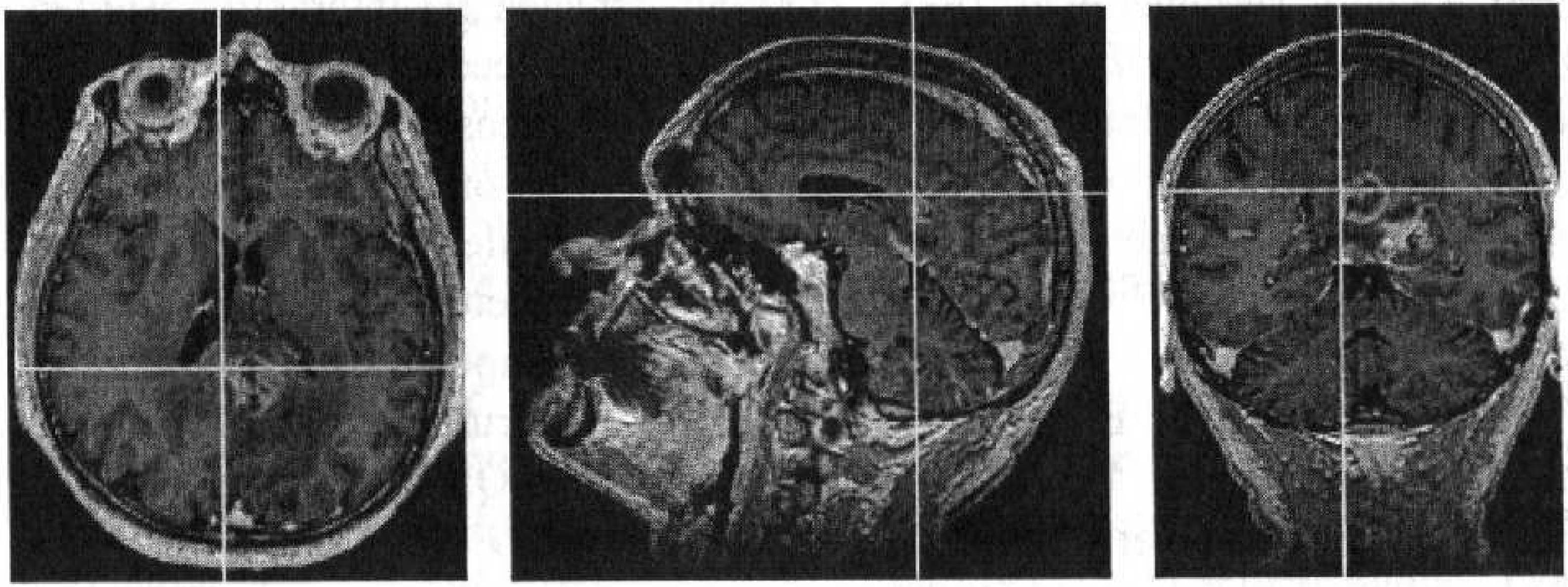
- die Planarisierungs- (Projektion auf eine Durchschnittsebene) und Triangulierungsverfahren sind für nichtplanare Polygone meist sehr zeitaufwendig und daher nicht effizient
- statt dessen kann durch ein rekursives Verfahren das nichtplanare Polygon sofort verarbeitet werden
- das Polygon wird dabei durch eine Teilungsebene, die durch zwei nicht unmittelbar aufeinanderfolgende Punkte definiert ist, getrennt
- die beiden resultierenden Hälften des Polygons werden auf die gleiche Weise getrennt, bis man nicht weiter teilbare Dreiecke erhält

Rekursives Triangulierungsverfahren für nichtplanare Ringe

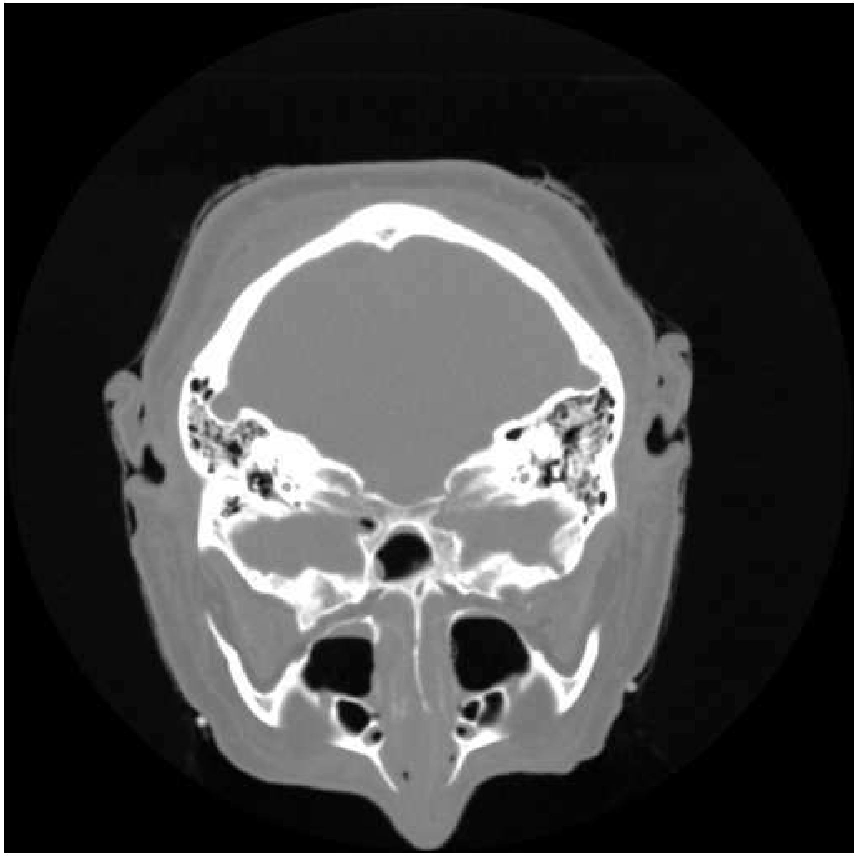


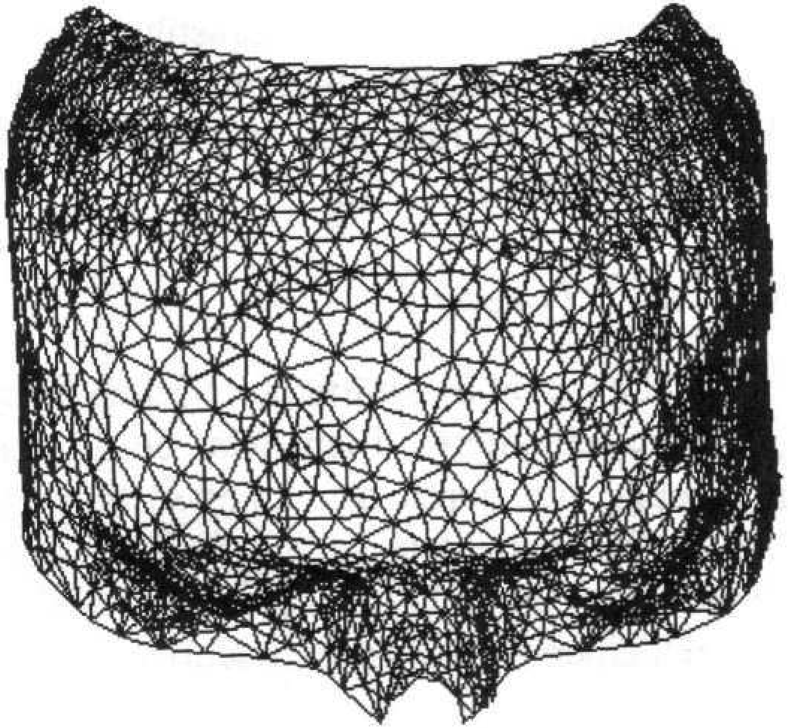
- zur Auswahl der Teilungsebene werden die Normalabstände aller Punkte zur Teilungsebene berechnet
- das Minimum der Normalabstände ist dann das Kriterium zur Auswahl der Teilungsebene
- der letzte Schritt beim Marching-Cube-Algorithmus ist die Berechnung der Normalenvektoren für jeden Eckpunkt der Dreiecke
- diese werden für die Beleuchtung und Schattierung eines 3D-Modells benötigt

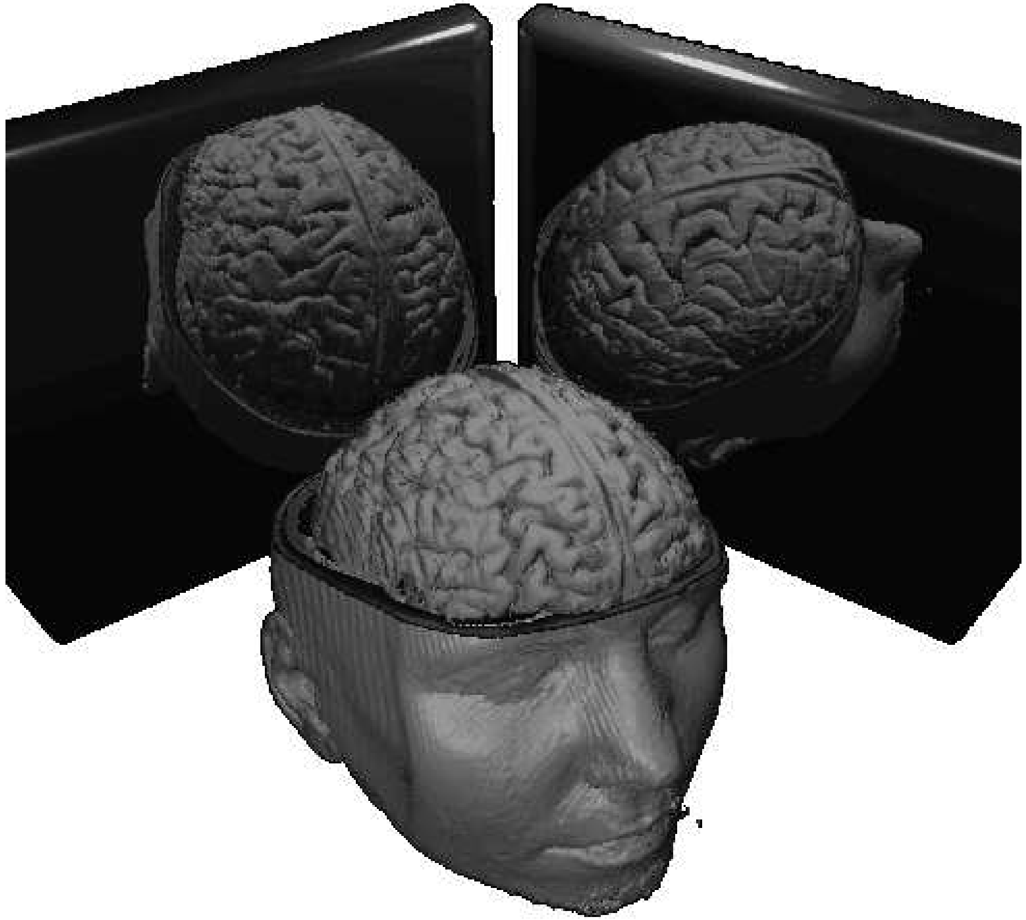
7 Impressionen



axial sagittal coronar







8 Quellenangaben

- http://people.freenet.de/hmueller/marchingCube.html
- http://wwwipr.ira.uka.de/~megi/SEMINAR/WS_00_01/indexpro.html
- http://kom.auc.dk/~zeek/kowd/mcubes/ind.html
- http://www.cosc.brocku.ca/Project/info/grossi/detmarch.html
- http://www.iue.tuwien.ac.at/publications/PhD%20Theses/strasser/node45.html
- http://weblogs.medien.uni-weimar.de:8080/vg/Marching%20Cube.pdf

- http://www.csit.fsu.edu/~erlebach/course/sciviz_f2001/marching_cubes.html
- http://www.cs.princeton.edu/courses/archive/spring00/cs5987b/lectures/voxels/sld011.htm