

# 3D-Visualisierung

Visual Computing

Winter Semester 2023-2024



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

**Prof. Dr. A. Kuijper**

Mathematical and Applied Visual Computing (MAVC)

Graphisch-Interaktive Systeme (GRIS)

Fraunhofer IGD

Fraunhoferstrasse 5

D - 64283 Darmstadt

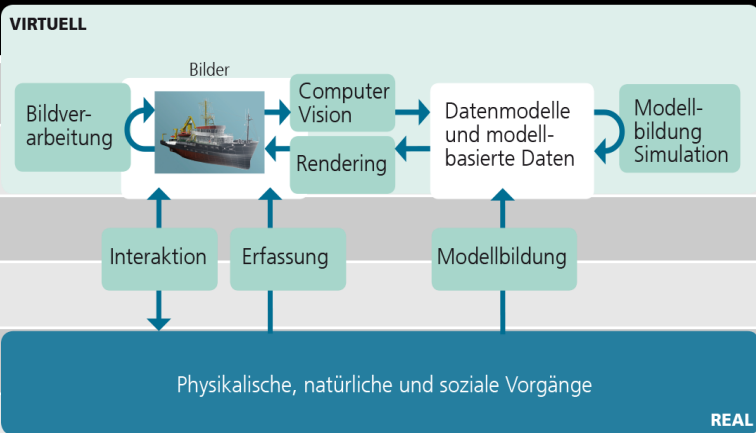
E-Mail: [office@gris.tu-darmstadt.de](mailto:office@gris.tu-darmstadt.de)

<http://www.gris.tu-darmstadt.de>

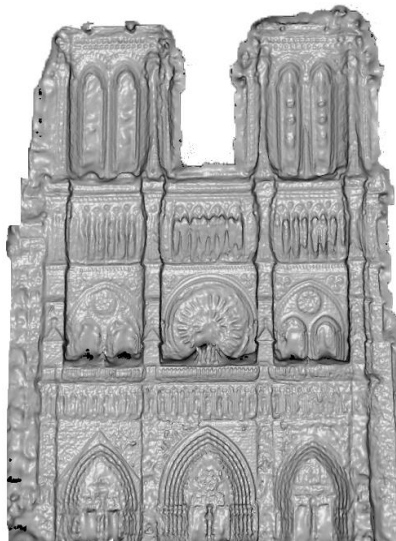
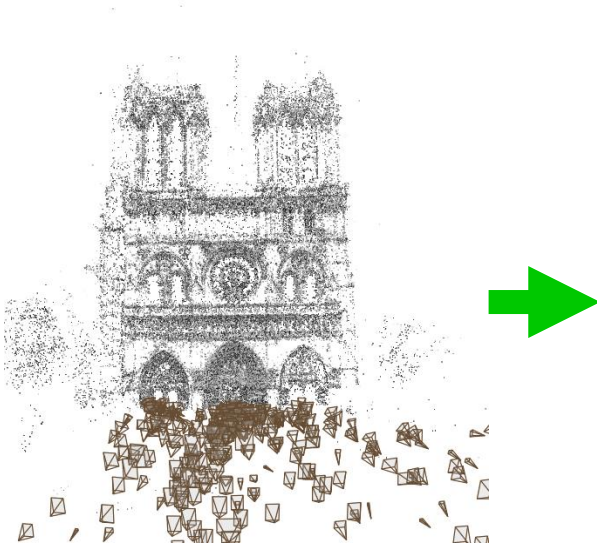
<https://www.mavc.tu-darmstadt.de>

---

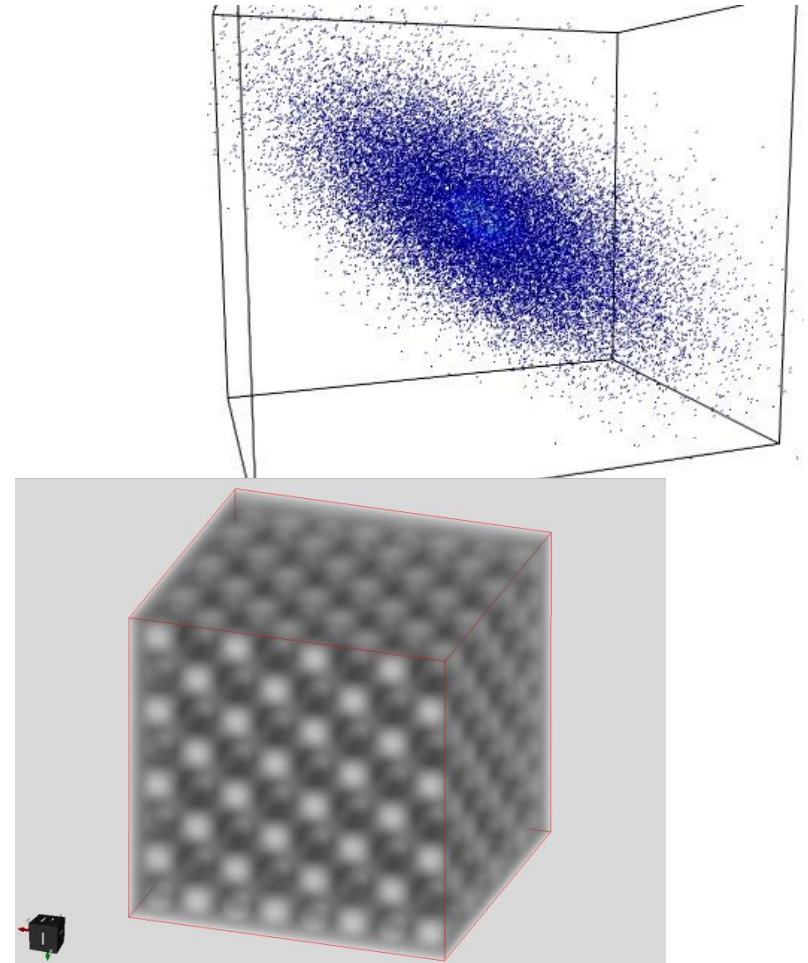
# Semesterplan

Datum		
20. Okt	Einführung + <a href="#">Visual Computing</a>	
27. Okt	<a href="#">Wahrnehmung</a>	
03. Nov	<a href="#">Objekterkennung</a> und <a href="#">Bayes</a>	
10. Nov	<a href="#">Fourier Theorie</a>	
17. Nov	<a href="#">Bilder</a>	
24. Nov	<a href="#">Bildverarbeitung</a>	
01. Dez	<a href="#">Grafikpipeline</a> & <a href="#">Eingabemodalitäten</a> & <a href="#">VR+AR</a>	
08. Dez	<a href="#">Transformationen</a> & <a href="#">2D/3D Ausgabe</a>	
<b>15. Dez</b>	<a href="#">3D-Visualisierung</a>	
12. Jan	X3D – 3D in HTML	
19. Jan	Informationsvisualisierung	
26. Jan	User Interfaces + Multimedia Retrieval	
02. Feb	Farbe	
09. Feb	Biometrie (?)	

- **3D-Daten**
- Triangulation von Punktwolken
- Indirekte Volumenvisualisierung
- Direkte Volumenvisualisierung



- Messwerte werden im dreidimensionalen Raum verteilt
- Jeder Wert hat drei Koordinaten  $(x, y, z)$
- Werte können gleichmäßig oder ungleichmäßig verteilt werden
- Ein Messwert kann skalar oder höherdimensional sein (z.B. ein Vektor)



# Gewinnung von 3D-Daten Terrain

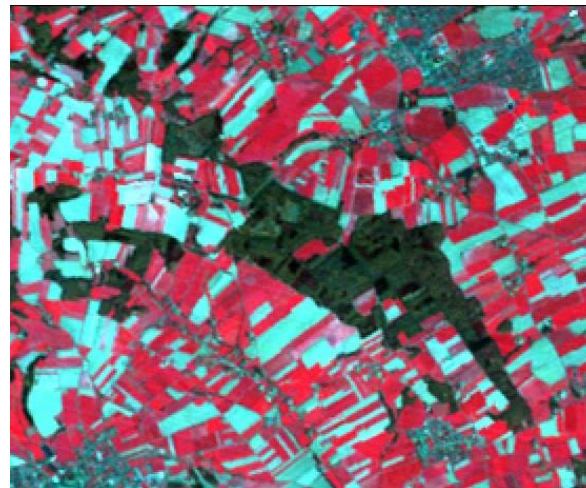
- An beliebigen Positionen (x, y) eines Terrains wird die Höhe z gemessen

→ 3D-Position



- Oberflächeninformationen (z.B. Vegetation) werden mithilfe von Satellitenbildern gewonnen

→ Skalarer Wert

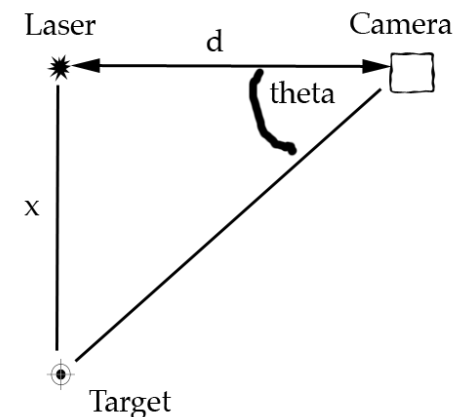




# Gewinnung von 3D-Daten

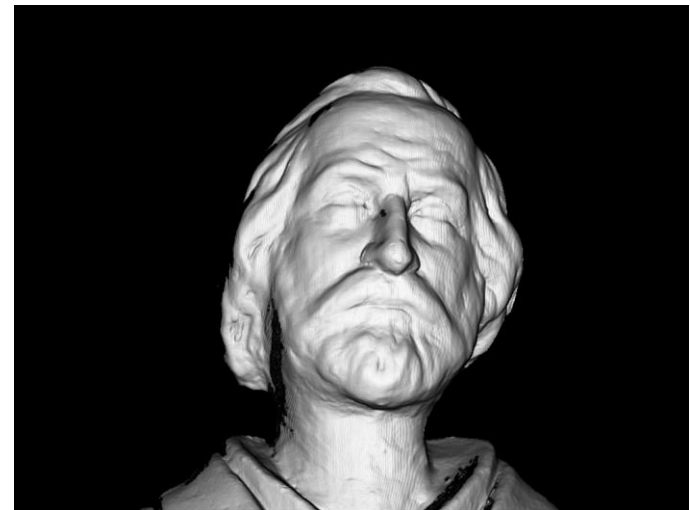
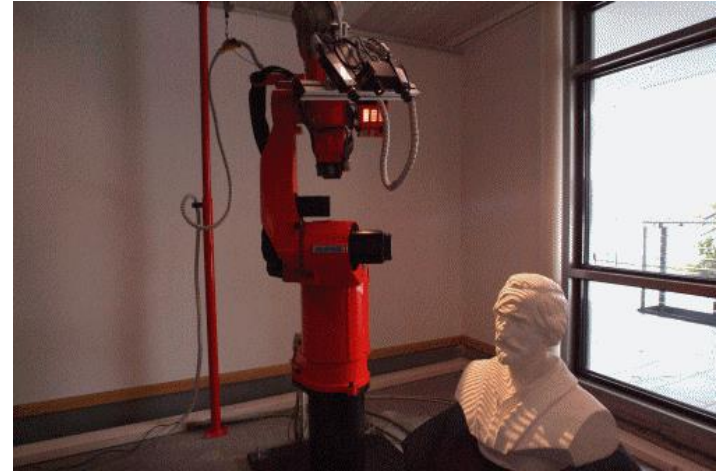
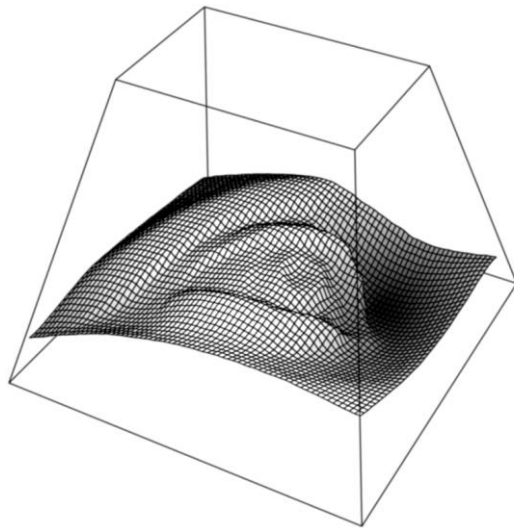
## Laser Scanning

- Ein Laserstrahl wird auf die Oberfläche projiziert
- Triangulation
  - Die Distanz zwischen Laser und Kamera ist bekannt
  - Der Winkel des Lasers ist bekannt
  - Der Winkel, in dem der Laserpunkt gesehen wird, wird gemessen
- ➔ Die Distanz von diesem Punkt kann abgeleitet werden
- ➔ Unstrukturierte Punktwolke



# Gewinnung von 3D-Daten Range Images

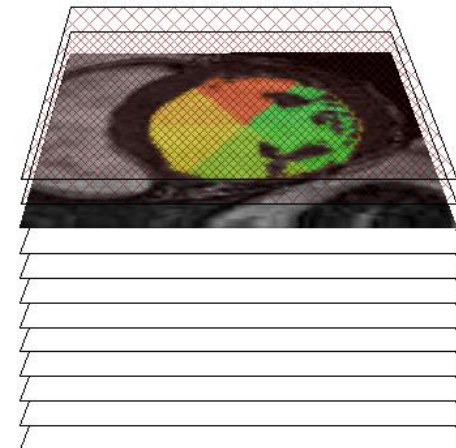
- Range Image  $r(u,v)$  →  
Tiefeninformationen
- Pixelinformationen als 3D-Punkt  
 $(u, v, r(u,v))$



# Gewinnung von 3D-Daten

## Medizinische Bilddaten

- Bildgebende Geräte um physikalische Eigenschaften zu messen:
  - Protonenkonzentrationen (MRI)
  - Gewebedichte (CT)
  - ‚Dichte‘ (Ultraschall)
  - ...
- Man erhält einen Stapel von parallelen, manchmal gleich weit voneinander entfernten Scheiben (Slices)
  - reguläres 2+1D-Gitter

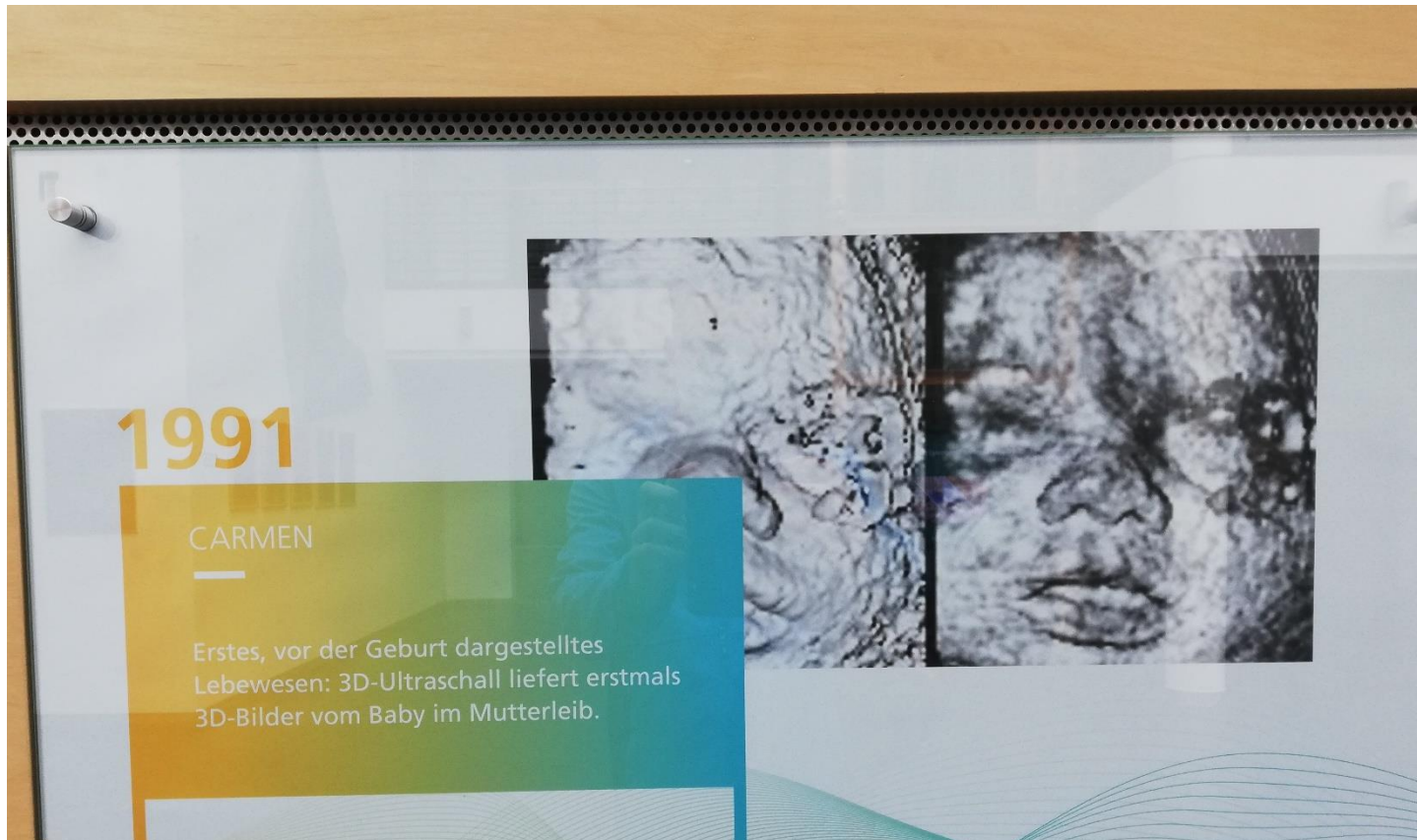




# Carmen



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT



Tucan:

- **20-00-0155-iv Bildverarbeitung**
- **20-00-0379-vI Medizinische Bildverarbeitung**

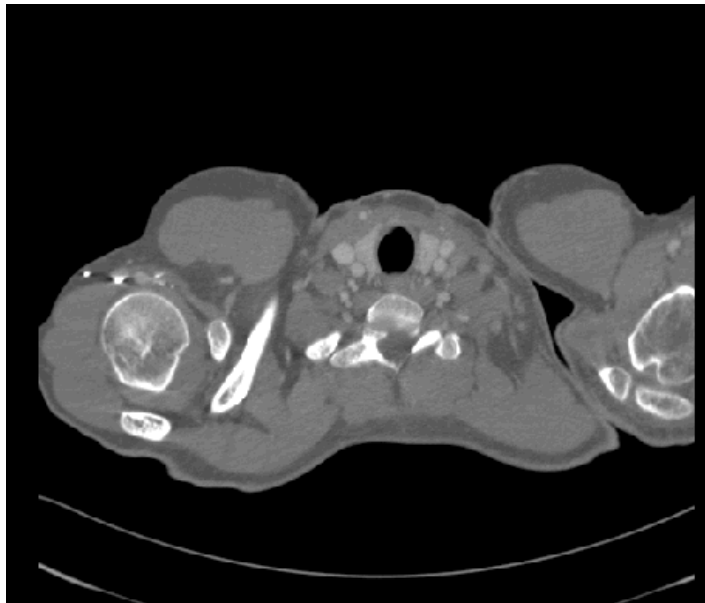
- <https://www.aerzteblatt.de/nachrichten/119505/Ultraschall-als-Babyfernsehen-wird-ab-2021-verboten>
- Im Gesetzestext heißt es: „Bei der Anwendung von Ultraschallgeräten zu nicht-medizinischen Zwecken darf ein Fötus nicht exponiert werden.“  
Begründet wird dies damit, dass es sich bei dem Fötus um eine schutzbe-  
fohlene Person handele, die der Untersuchung und den damit  
verbundenen möglichen Nebenwirkungen nicht zustimmen kann und  
selber keinen Nutzen aus der Untersuchung zieht.



# Gewinnung von 3D-Daten

## Beispiel: Computertomographie (CT)

Riesige Datenmengen...



1220 Slices



# Gewinnung von 3D-Daten

## Beispiel: Schneiden

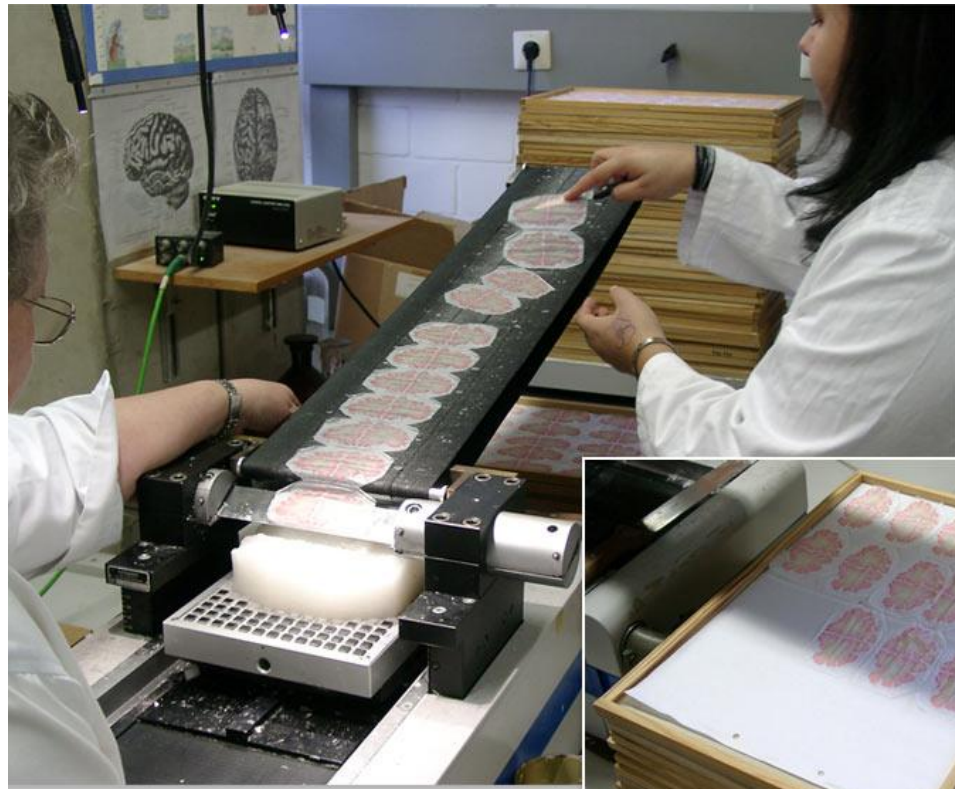
- [BigBrain Atlas Unveiled:](#)

Science 21 June 2013:

Vol. 340 no. 6139

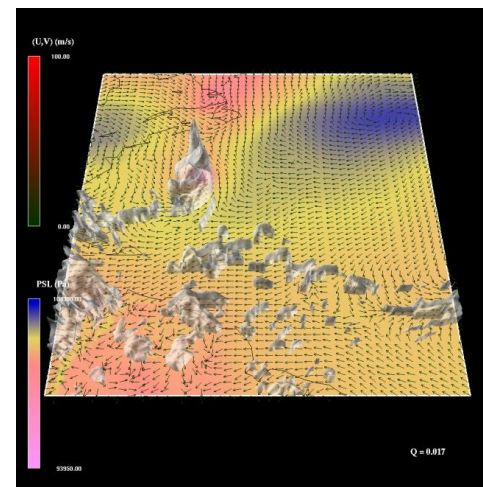
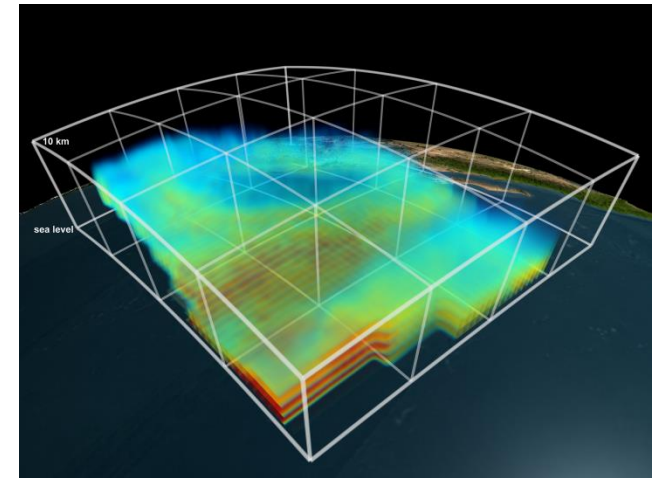
pp. 1472-1475

- 7404 slices....



# Gewinnung von 3D-Daten Wetter

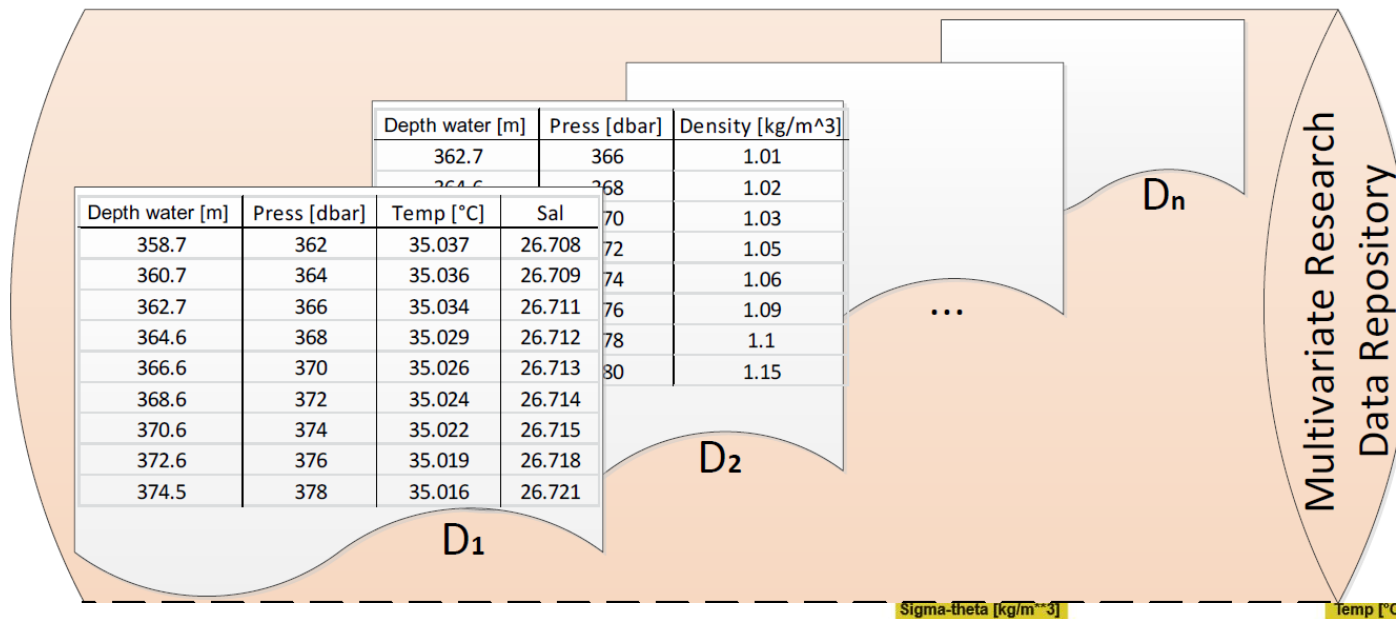
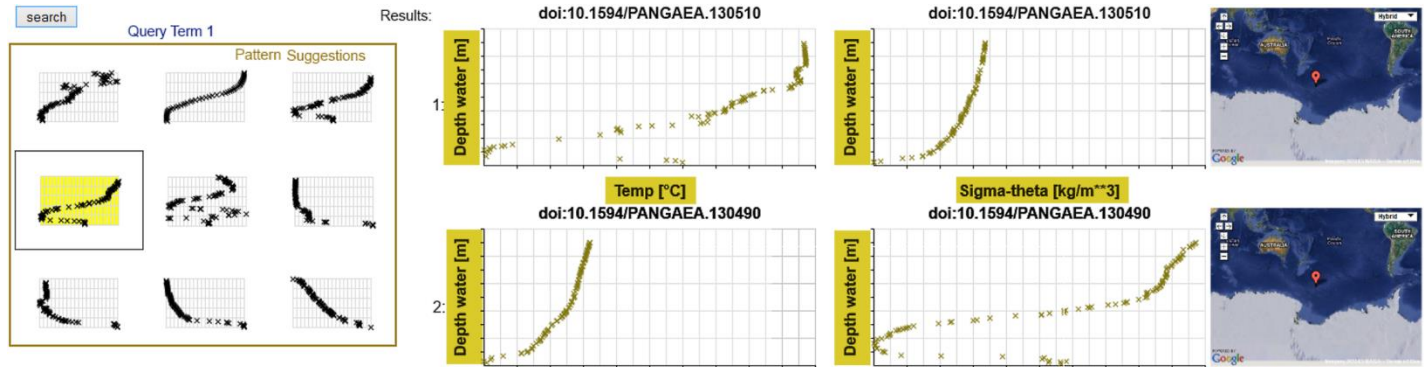
- Wetterparameter werden für bestimmte Regionen auf verschiedenen Höhen gemessen und simuliert:
  - Temperatur
  - Barometerstand
  - Niederschlag
  - Windrichtung
  - ...
- ➔ Zellen einer bestimmten Größe (einige Kilometer) in einem regulären Raster
- Skalare oder vektorielle Daten





# Gewinnung von 3D-Daten Klimaforschung

Vielfaltige  
Messdaten:  
(x,y,z,T,p, $\rho$ ,...)

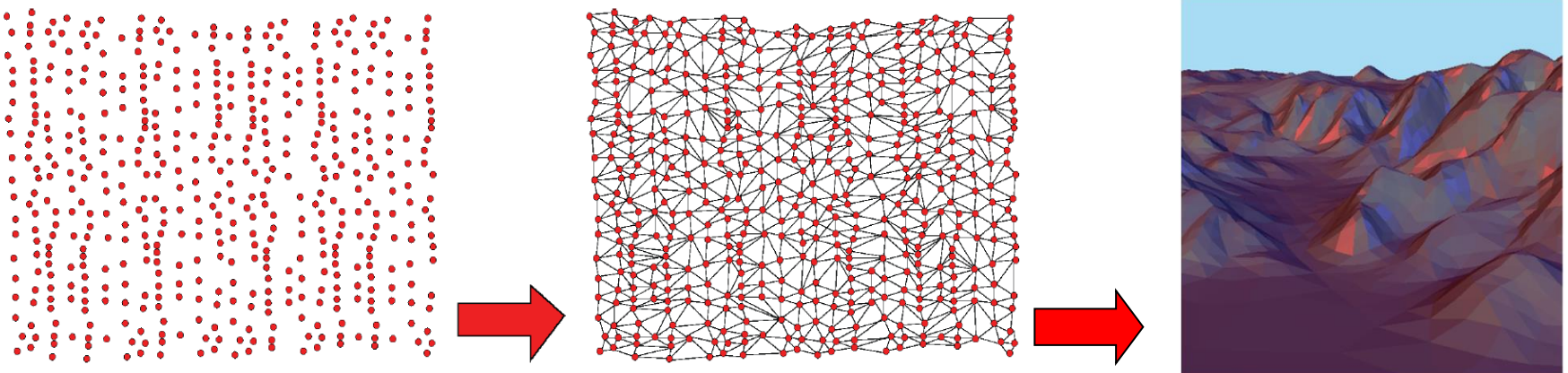


Total Result Count: 12

- 3D-Daten
- **Triangulation von Punktwolken**
- Indirekte Volumenvisualisierung
- Direkte Volumenvisualisierung

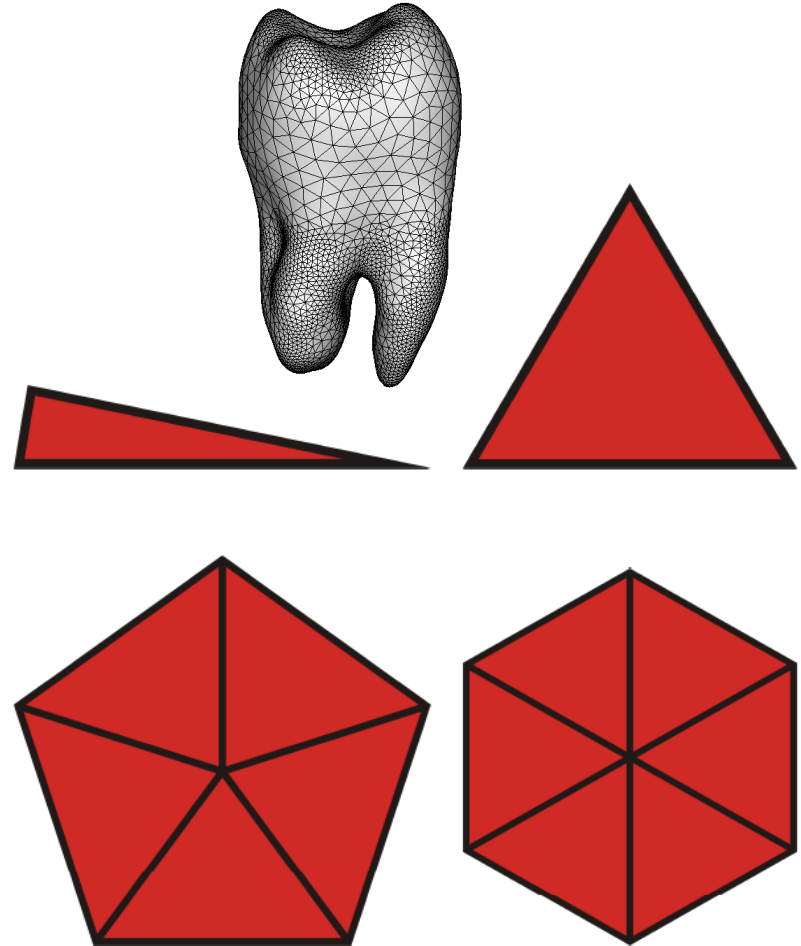
# Triangulation von Punktwolken

- Unstrukturierte Punktmenge  $s_i = (x_i, y_i, z_i)$  auf der Oberfläche  $S$   
→ Punktwolke
- Für „einfache“ Oberflächen (ohne Falten) gilt:
  - Punkte können auf eine Ebene projiziert ( $s_i \Rightarrow \underline{s}_i$ ) und in 2D trianguliert werden
    - Planare Triangulation
  - Das 2D-Dreiecksnetz wird dann entsprechend den Werten von  $z_i$  deformiert



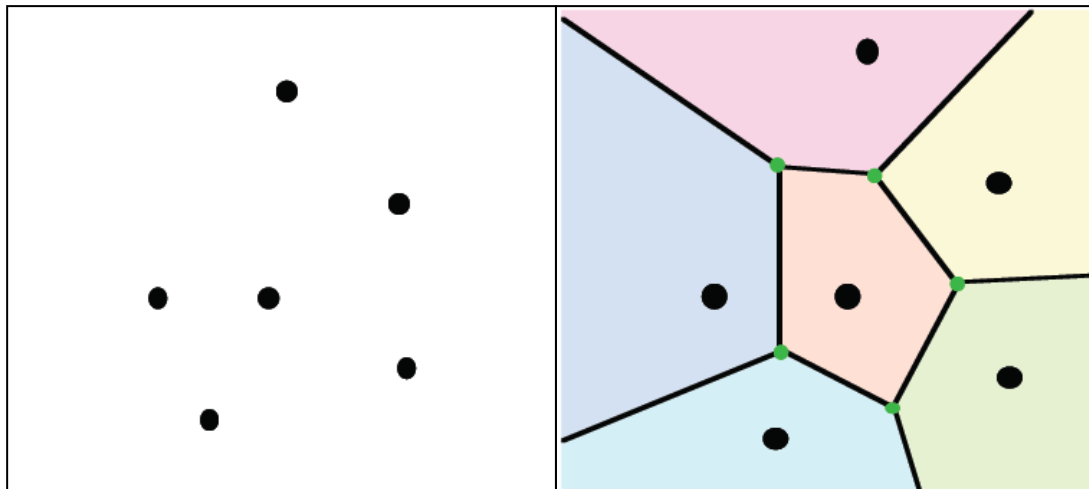
# Bevorzugte Triangulationen

- Dreiecksform und –winkel
  - Ideal (rechts):
    - Gleichseitig; alle Winkel sind  $60^\circ$
    - Knotengrad 6
- Vorteile:
  - Aussehen
  - Numerische Stabilität
  - Post-Processing (Unterteilung)



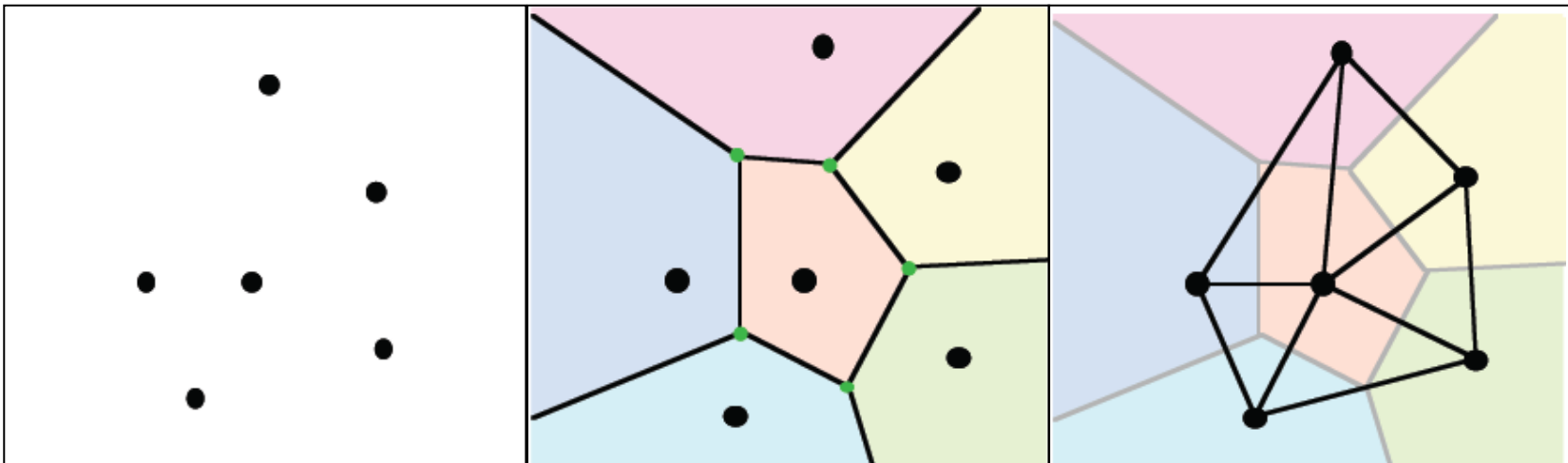
# Voronoi-Diagramm

- Für jeden (projizierten) Punkt  $\underline{s}_i$  kann eine Voronoi-Zelle definiert werden:
  - Eine solche enthält alle Punkte, die näher an  $\underline{s}_i$  als zu allen anderen Orten liegen
  - Kanten einer Voronoi-Zelle → Punkte mit zwei nächsten Punkten
  - Knoten einer Voronoi-Zelle → Punkte mit drei oder mehr nächsten Punkten
- Die Voronoi-Zellen „parkettieren“ die 2D-Fläche  
→ Voronoi-Parkettierung (Tesselation), Voronoi-Diagramm





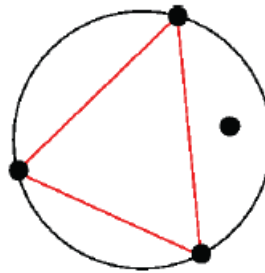
- Man betrachtet den dualen Graph eines Voronoi-Diagramms
    - Die Zentren von benachbarten Polygonen werden mit Linien verbunden, die zur jeweils gemeinsamen Kante orthogonal sind
- ➔ Delaunay-Triangulation (möglicherweise sind Korrekturen nötig)



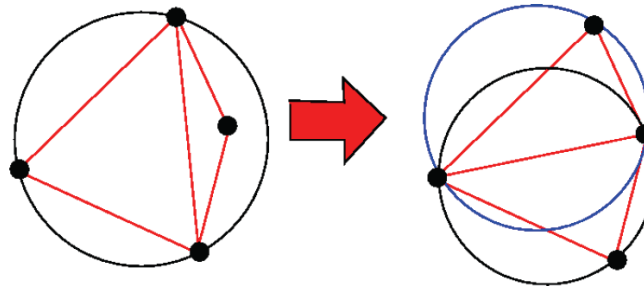
# Delaunay-Triangulation

- Ein Dreiecksnetz ist eine Delaunay-Triangulation, wenn alle Umkreise von allen Dreiecken im Netz leer sind

→ nicht erlaubt:



- Korrektur der Delaunay-Triangulation durch Umdrehen der Kanten („Edge Flipping“):



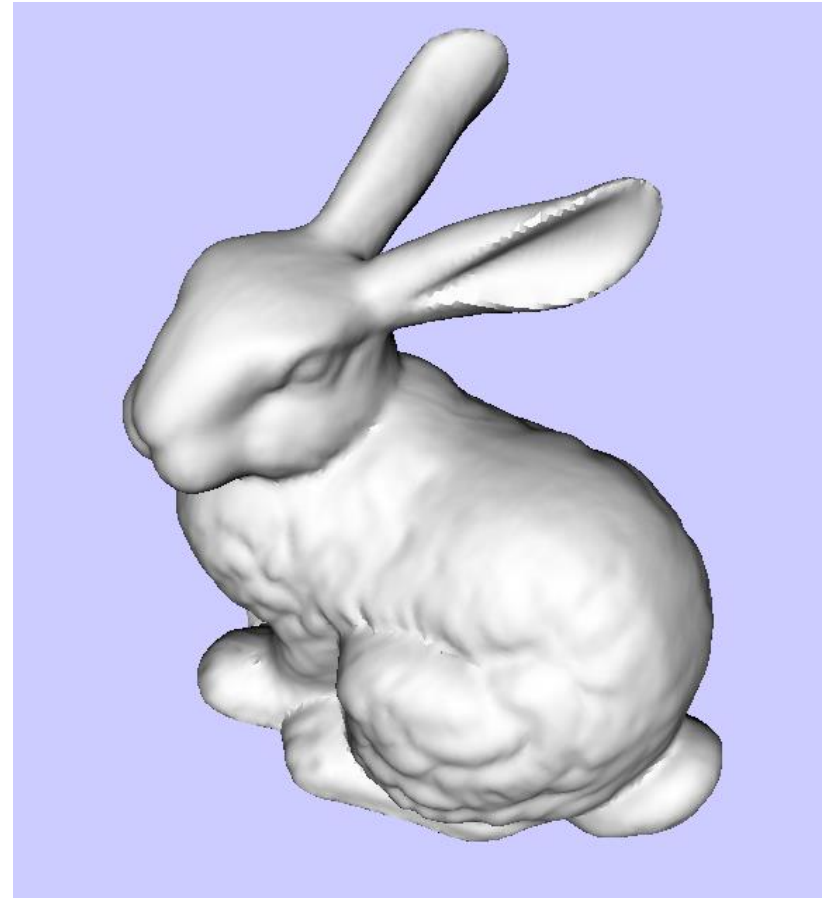
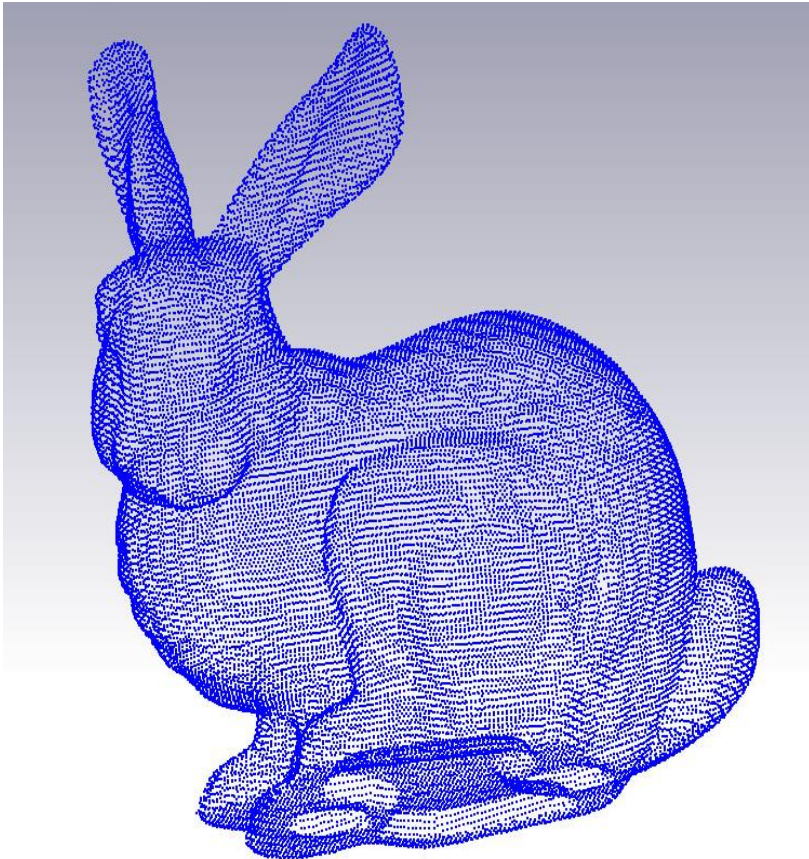


<http://www.pi6.fernuni-hagen.de/GeomLab/VoroUpdate/>

Backup, nur Voronoi: <http://alexbeutel.com/webgl/voronoi.html>

<https://cartography-playground.gitlab.io/playgrounds/triangulation-delaunay-voronoi-diagram/>

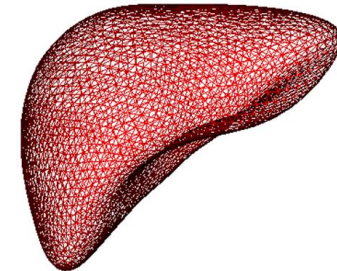
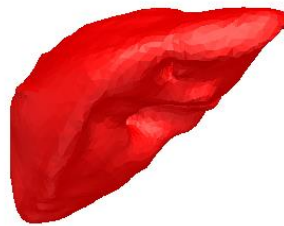
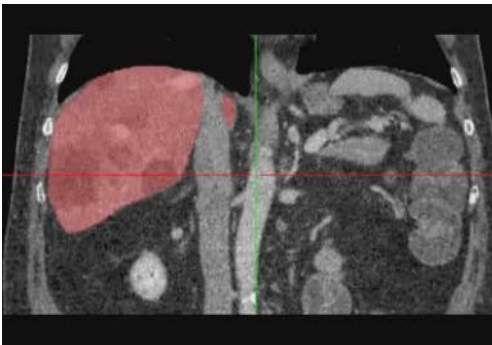
# Triangulation von Punktwolken



# Beispiel: Medizinische Modelle

- Modelle durch Segmentierung lernen  
→ Eine Punktwolke definiert die Umrisse eines Organs.
- Die gelernten Modelle werden verwendet, um neue Bilder zu segmentieren oder zu klassifizieren

(Dissertation M. Erdt)



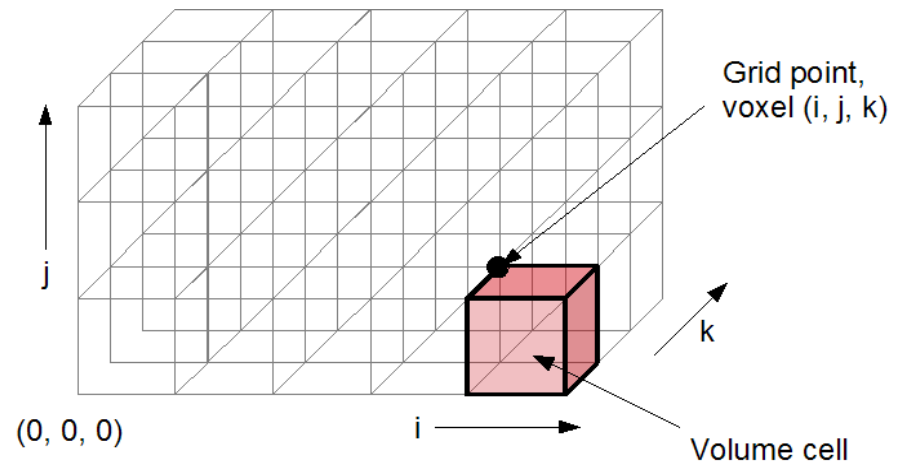




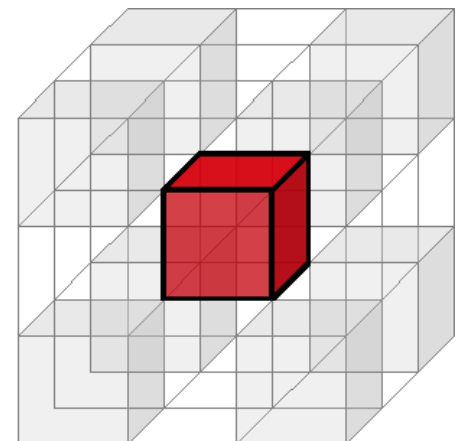
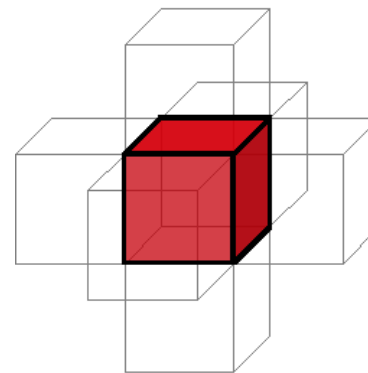
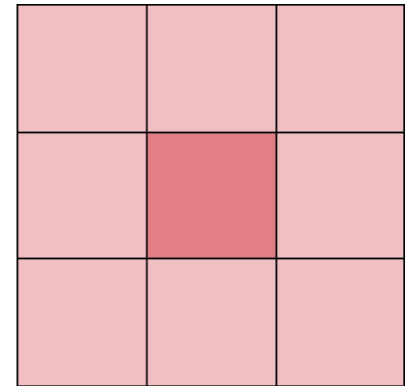
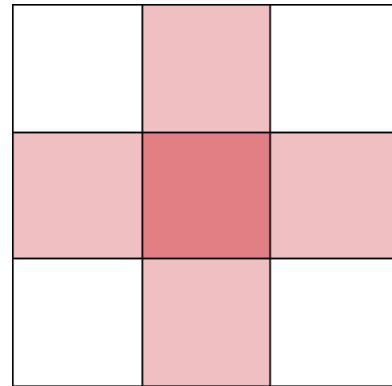
- 3D-Daten
- Triangulation von Punktwolken
- **Indirekte Volumenvisualisierung**
- Direkte Volumenvisualisierung



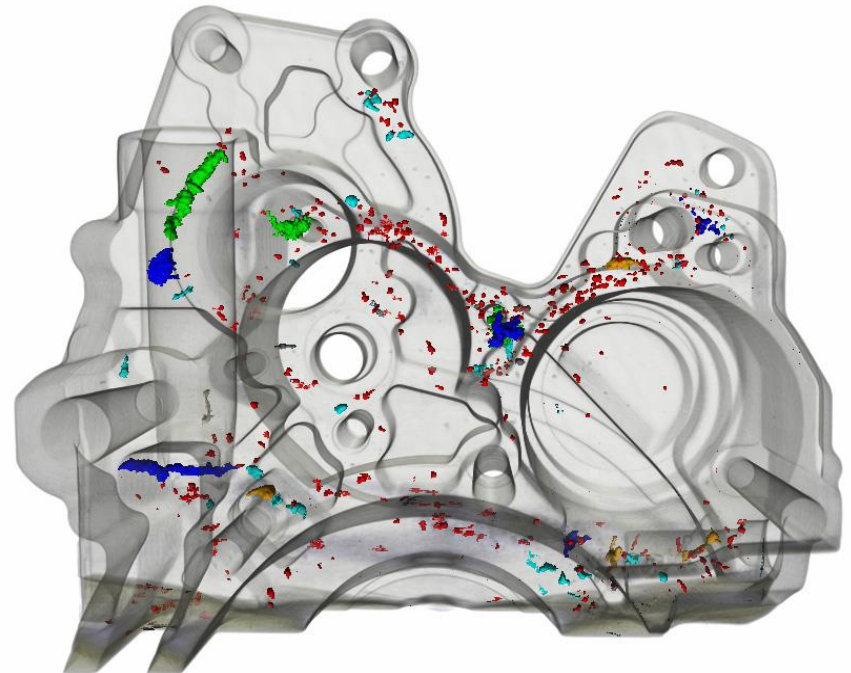
- Reguläres 3D-Raster
- Volumenelemente → Voxel
- Slice-Dicke:
  - Abstand in Slice-Auswahl-Richtung
  - Oftmals größer als Pixelabstände (anisotropische Volumen)
- Rasterposition: Index  $(i, j, k)$



- Voxel sind adjazent zu einem Referenzvoxel
- Im Zweidimensionalen:
  - Kanten: N4
  - Kanten + Ecken: N8
  - (N6: hexagonal)
- Im Dreidimensionalen:
  - Flächen: N6
  - Flächen + Kanten: N18
  - Flächen + Kanten + Ecken: N26



- Eine Menge von Volumendaten enthält viele Informationen
  - Eingeschränkte Geschwindigkeit beim Rendering
  - Verdeckung von hinteren Elementen durch weiter vorne befindliche.
- Man zeigt nicht das gesamte Volumen, sondern nur eine Teilmenge davon an
  - Oberflächenorientiert



Industrielle CT, 667x465x512 Voxel





- Trennung zwischen verschiedenen Strukturen → Eingrenzung
- Visualisierung dieser Eingrenzung → Erkennung der Strukturen
- Voxel an der Eingrenzung haben die gleiche Intensität → Isoflächen

- Isofläche:

- Implizite Fläche,  $V(x)$  Voxelwert,  $\tau$  Isowert

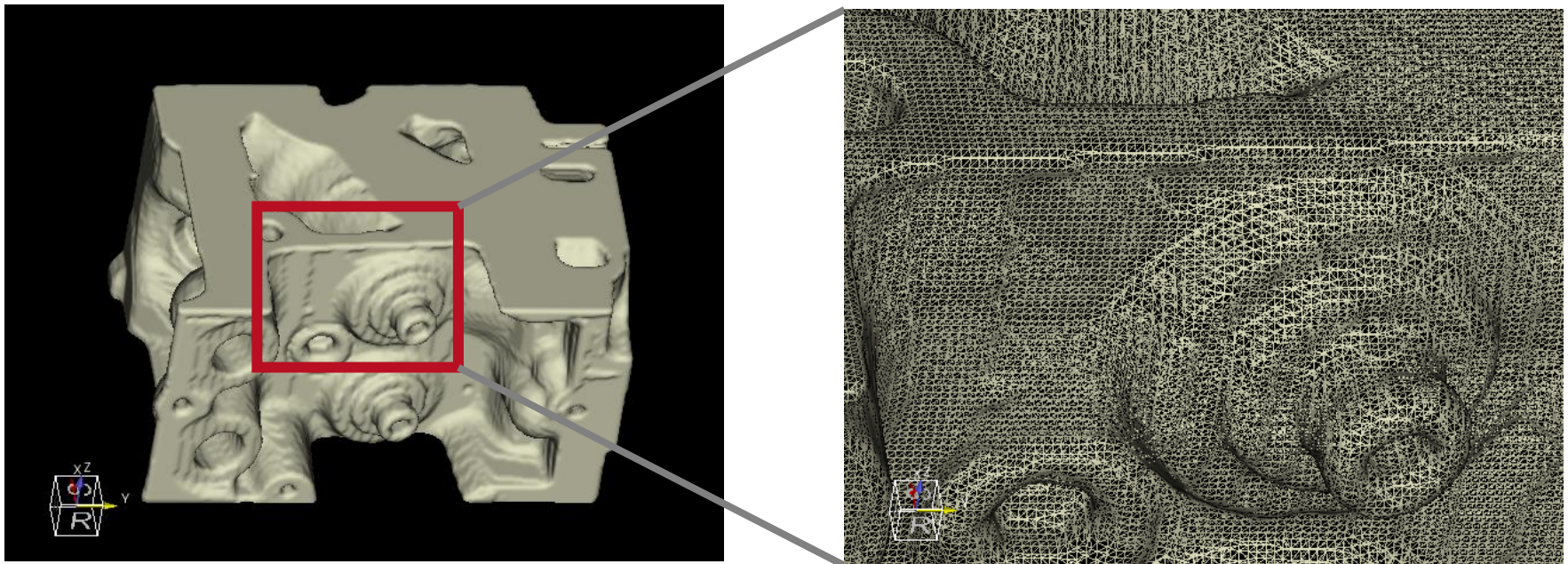


$$i(x) = V(x) - \tau = 0$$

- Aufteilung einer Datenmenge in  
innen ( $i(x) > 0$ ) und außen ( $i(x) < 0$ )
  - Definition eines Isowerts → Thresholding der Daten

# Polygon-Repräsentationen von Isoflächen

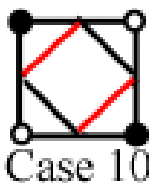
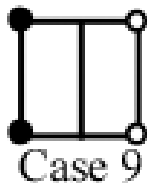
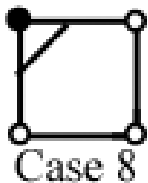
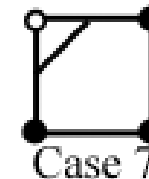
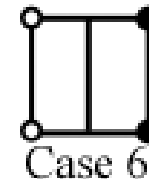
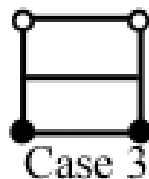
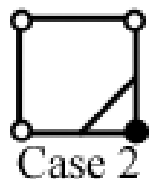
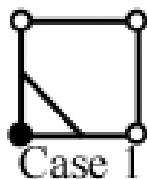
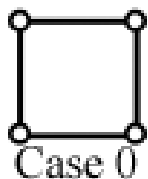
- Es wird ein Mesh berechnet, das die Oberfläche repräsentiert



# 2D: Marching Squares



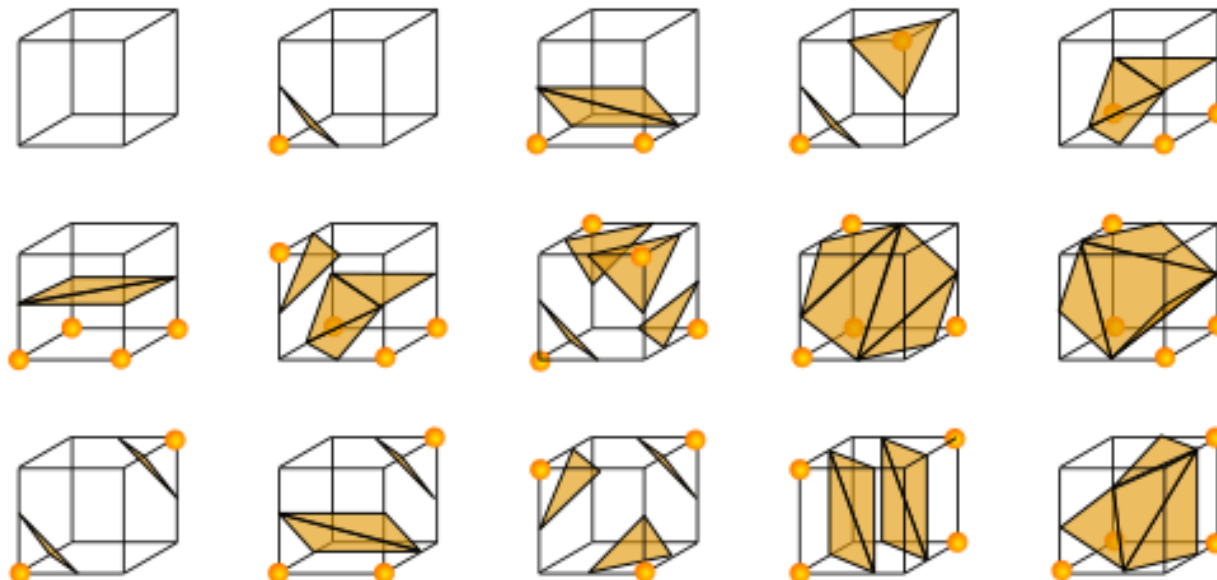
- Bildzellen seien durch ihre vier umgebenden Pixel definiert
- Pixel werden betrachtet, wenn sie größer oder gleich des Isowertes sind (d.h. Innerhalb oder auf der Begrenzung liegen) → „Set Pixel“-Zustand
- 16 mögliche Kombinationen „Set Pixel“ und „Reset Pixel“



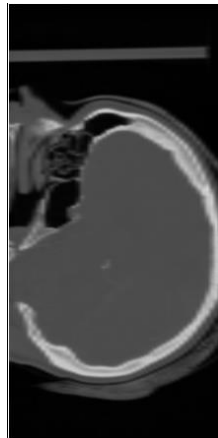
- 
- A 7x7 grid of points. A red polygon highlights a subset of points. The highlighted points are located at the following coordinates (row, column) starting from the top-left: (1, 2), (2, 3), (2, 4), (2, 5), (3, 3), (3, 4), (3, 5), (3, 6), (4, 3), (4, 4), (4, 5), (4, 6), (5, 3), (5, 4), (5, 5), (5, 6), (6, 3), (6, 4), (6, 5), (6, 6). The red polygon connects these points in a continuous path, starting from (1, 2) and ending at (6, 3).

# Marching Cubes

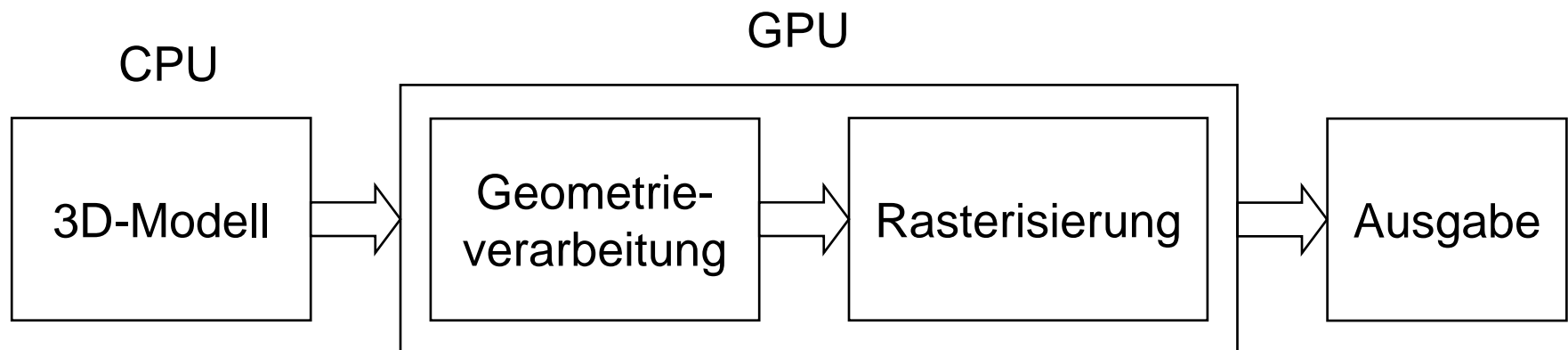
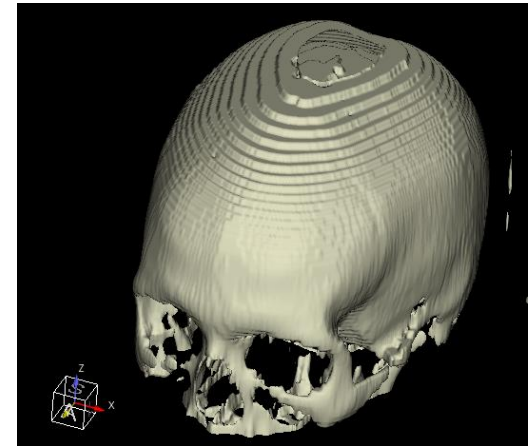
- „3D-Version“ der Marching Squares
- Eine Volumenzelle ist durch ihre acht umgebenden Voxel definiert
- 256 mögliche Kombinationen „Set Pixel“ und „Reset Pixel“
  - ➔ beachtet man die Symmetrie, erhält man 15 äquivalente Klassen



# Oberflächenrendering von Volumendaten



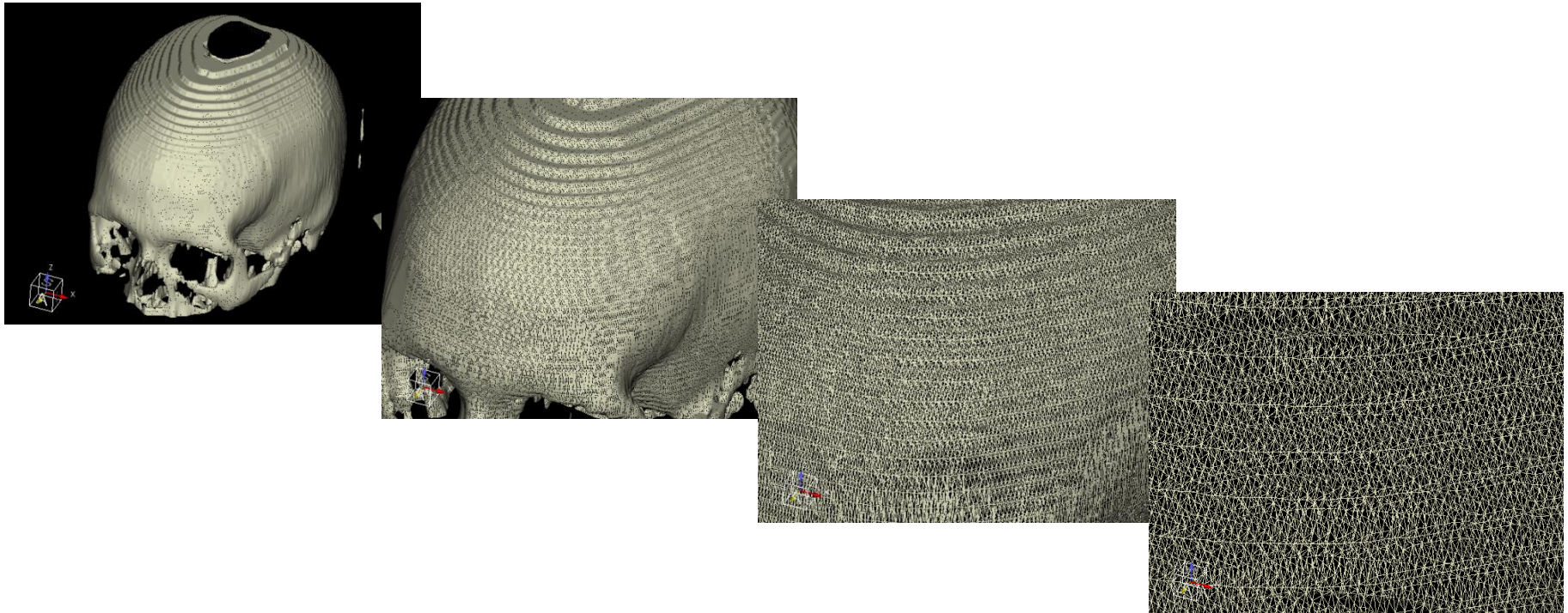
Thresholding +  
Marching Cubes



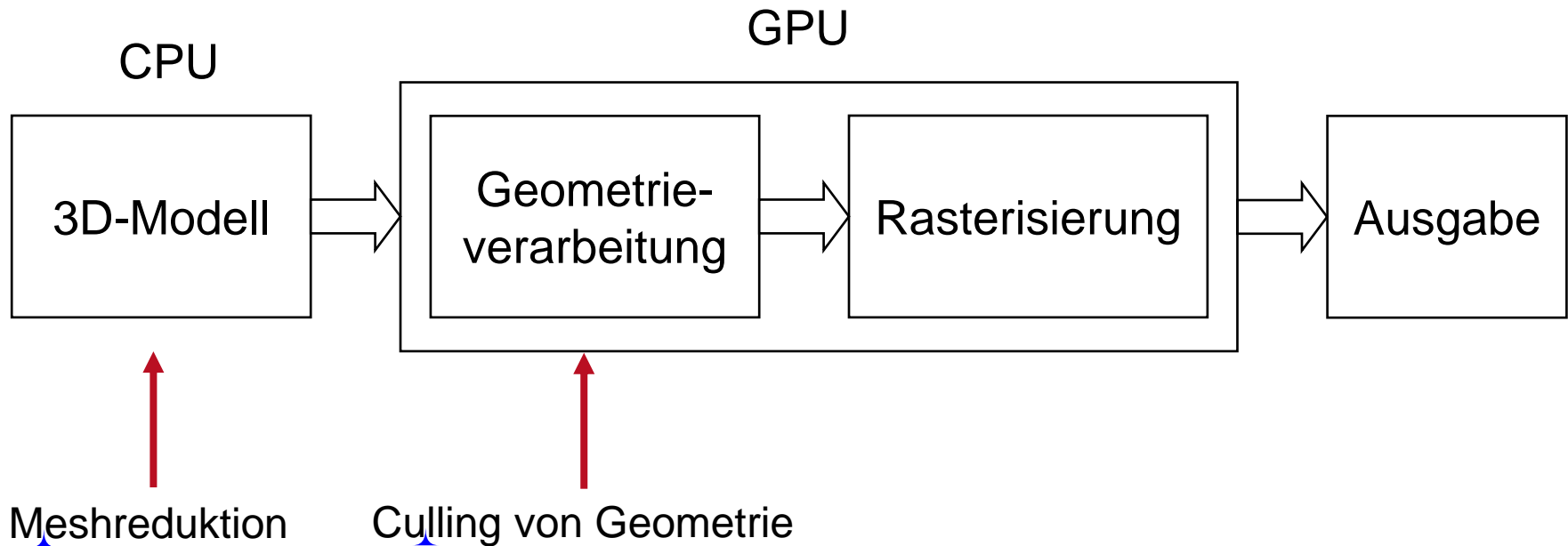




- Marching Cubes gibt viele Millionen Dreiecke aus
- Hoher Berechnungsaufwand für das grafische Subsystem



# Verbessern der Rendering Performance

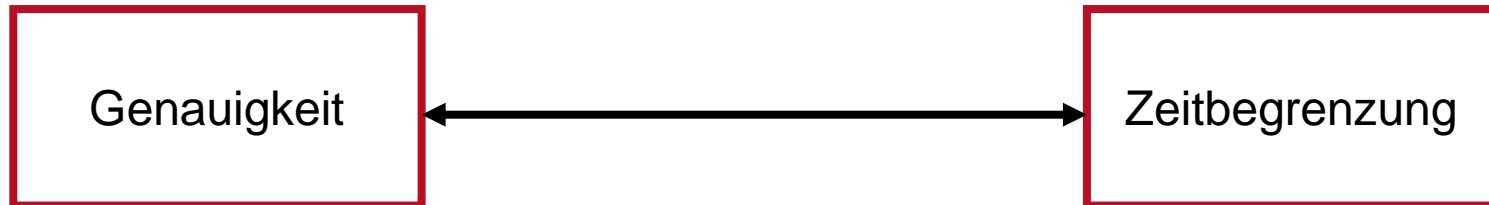




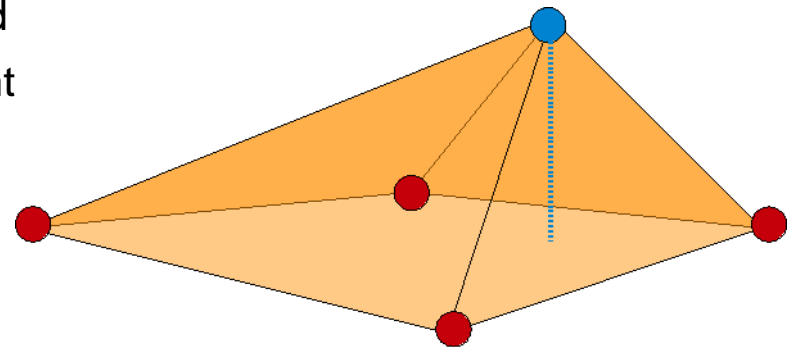
- Unsichtbare Polygone werden aus der Rendering-Pipeline entfernt.
  - **Backface-Culling:** Zum Betrachter gerichtete Rückseiten werden nicht gezeichnet, d.h. solche Polygone, deren Normalen vom Sichtpunkt wegzeigen.
  - **View-Frustum-Culling:** Polygone, die sich ganz oder teilweise außerhalb des View-Frustums befinden, werden nicht oder nur teilweise gezeichnet.
  - **Occlusion-Culling:** Polygone werden nach Tiefe sortiert und nur gerendert, wenn sie nicht vollständig durch andere verdeckt werden.  
Transparenz muss hier unbedingt beachtet werden!
  - ...



- Die Gesamtzahl an zu rendernden Polygonen wird verringert
- Die Akzeptanz der Vereinfachung ist abhängig vom Szenario



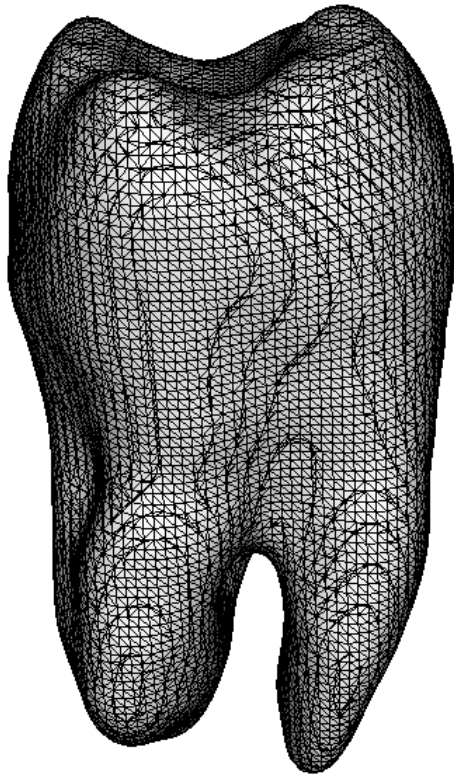
- Mögliche Differenzmetrik:
  - Die Distanz eines Vertex zur Ebene wird durch die benachbarten Vertices geformt



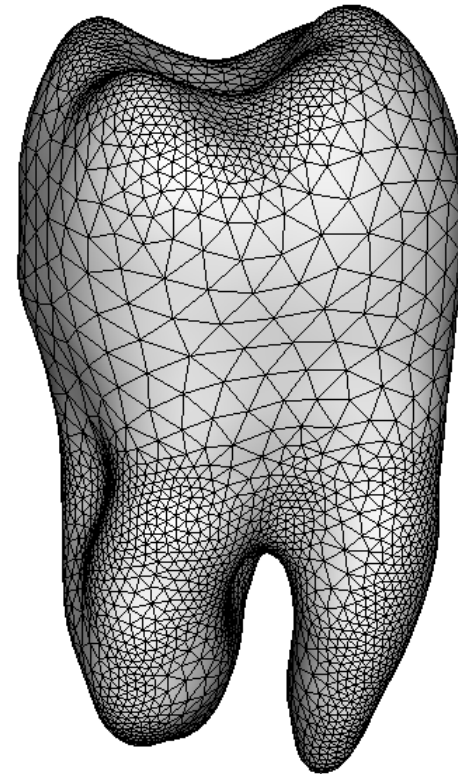
# Meshreduktion



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT



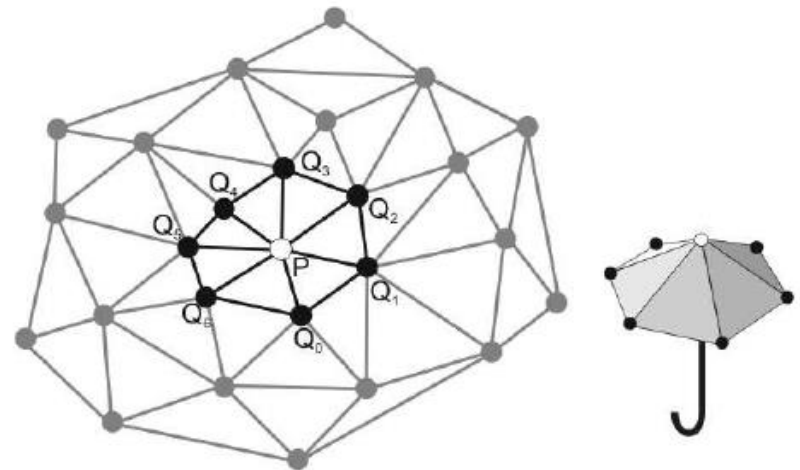
**Originalmesh:**  
**~24k Dreiecke**



**Neu abgetastetes  
Mesh: ~8k Dreiecke**



- Ziel: Gute Visualisierungen bereitstellen
  - Artefakte reduzieren
  - Löcher entfernen
- Herausforderung: Volumen erhalten
  - ➔ quantitative Informationen könnten benötigt werden
- Laplacesche Glättung
  - Man betrachtet eine „Regenschirm-Region“
  - Reduzierung hochfrequenter Oberflächen-informationen
    - ➔ Reduktion von Krümmungen

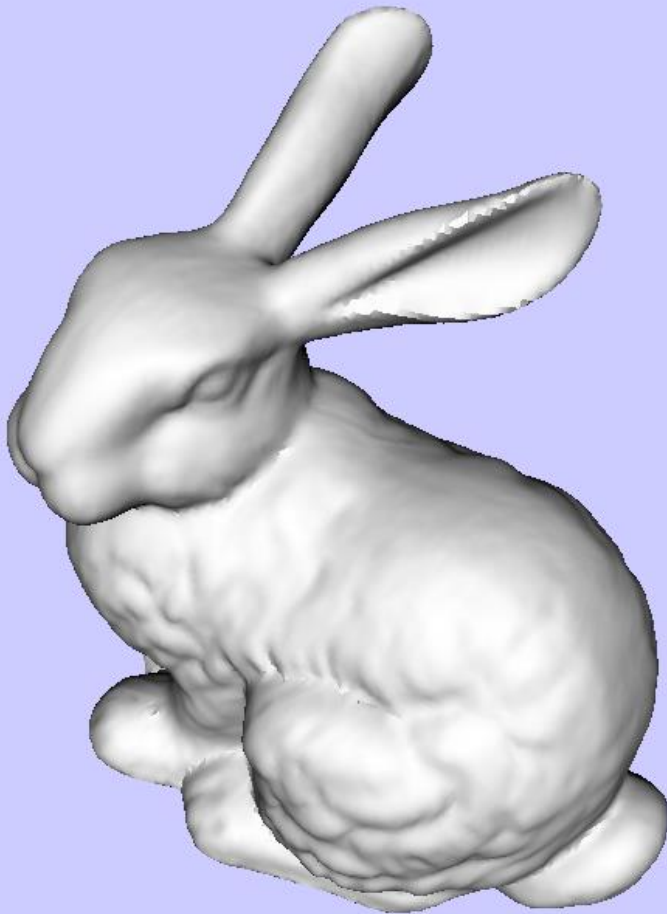




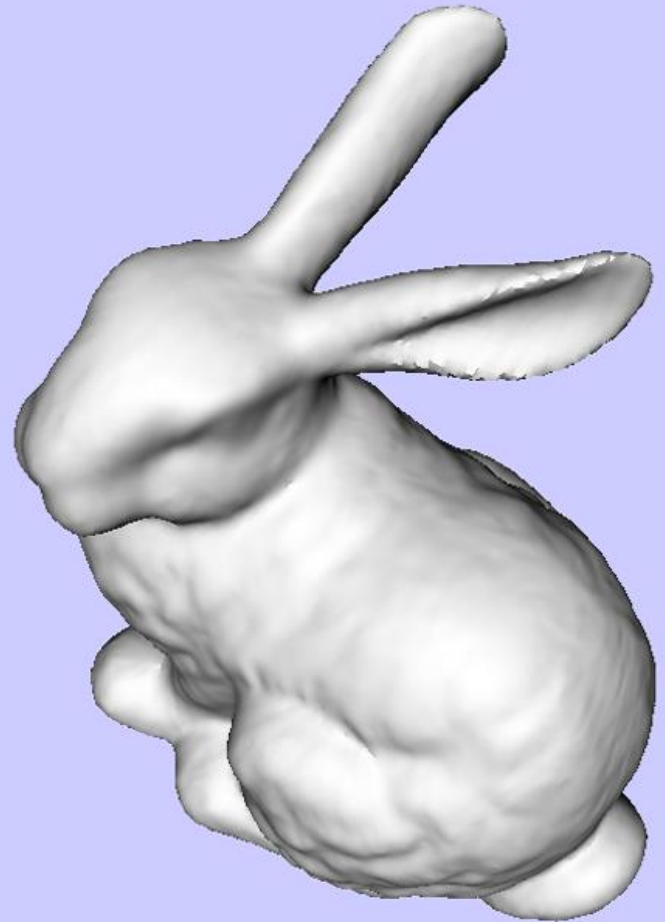
# Mesh-Glättung



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT



Original

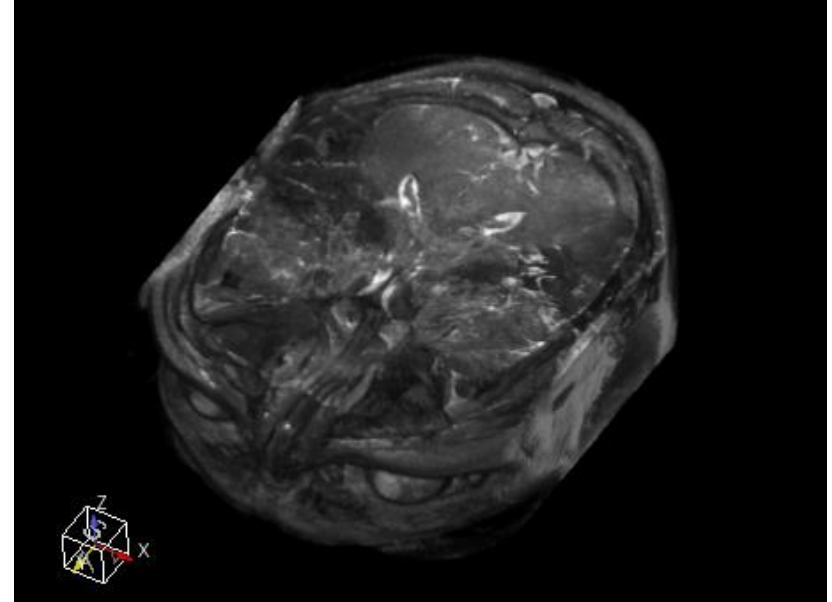
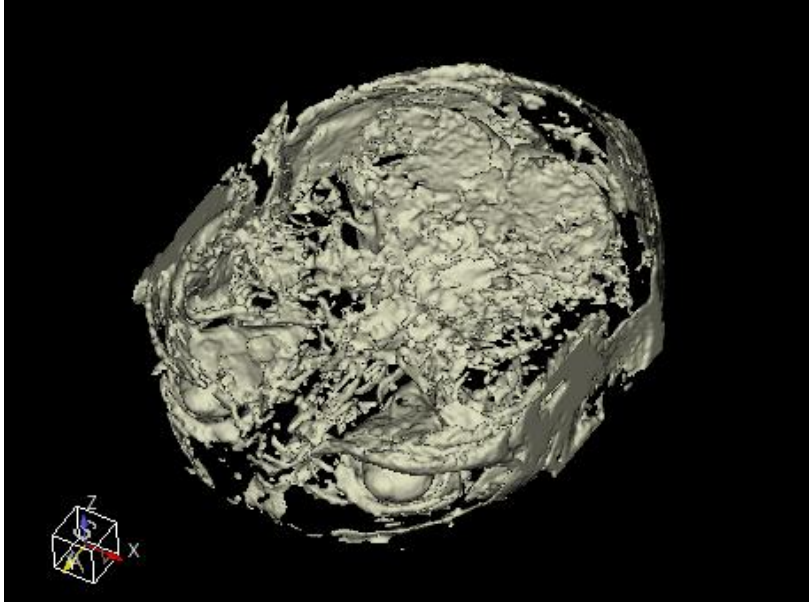


Geglättet

- 3D-Daten
- Triangulation von Punktwolken
- Indirekte Volumenvisualisierung
- **Direkte Volumenvisualisierung**

# Direkte Volumenvisualisierung

- *Indirekt:* Generierung einer Zwischendarstellung des Volumens
- Die Komplexität hängt von der Anzahl an Polygonen ab, z.B. Marching Cubes:
- *Direkt:* Visualisierung ohne Generierung einer Metadarstellung
- Die Komplexität hängt von der Anzahl der Voxel und der Auflösung der Anzeigefläche ab



# Modell für Volumenrendering: Density Emitter Model

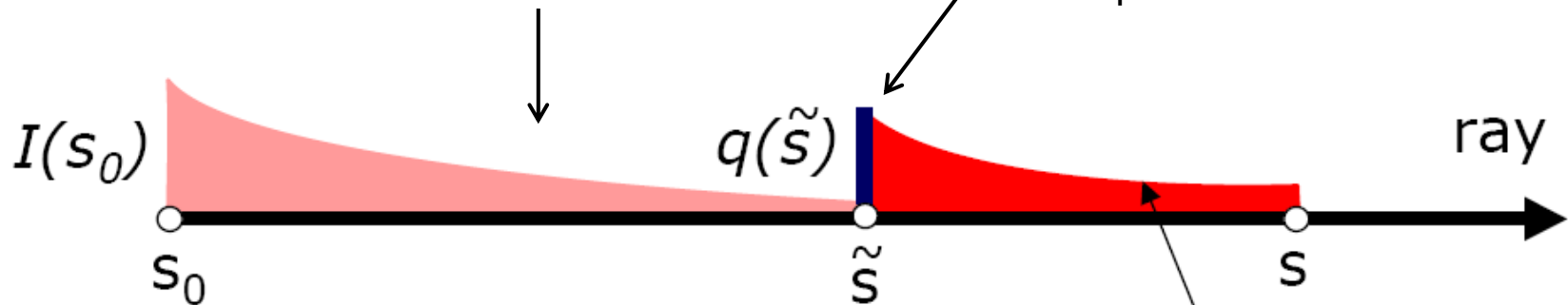
- Man betrachte nur Emission und Absorption
- Jeder Voxel in der Datenmenge ist eine kleine Lichtquelle
- Das Licht wird schwächer während es durch die Volumendatenmenge wandert
- Das Medium ist eine homogene Dichtewolke



# Volumen-Rendering-Gleichung

Das Licht wird schwächer während es durch die Volumendatenmenge wandert („Optische Tiefe“)

Jeder Voxel in der Datenmenge ist eine kleine Lichtquelle



Die Bestrahlungsstärke  $I$  des Hintergrundes  $s_0$  exponentiell abgeschwächt durch das Volumen plus exponentiell abgeschwächte neue Lichtquellen  $Q$ :

Active Emission at  $\tilde{s}$

Absorption along the distance  $\tilde{s} - s$

$$I(s) = I_{s_0} \cdot \exp\left(-\int_{s_0}^s \tau(t) dt\right) + \int_{s_0}^s Q(\tilde{s}) \cdot \exp\left(-\int_{\tilde{s}}^s \tau(t) dt\right) d\tilde{s}$$

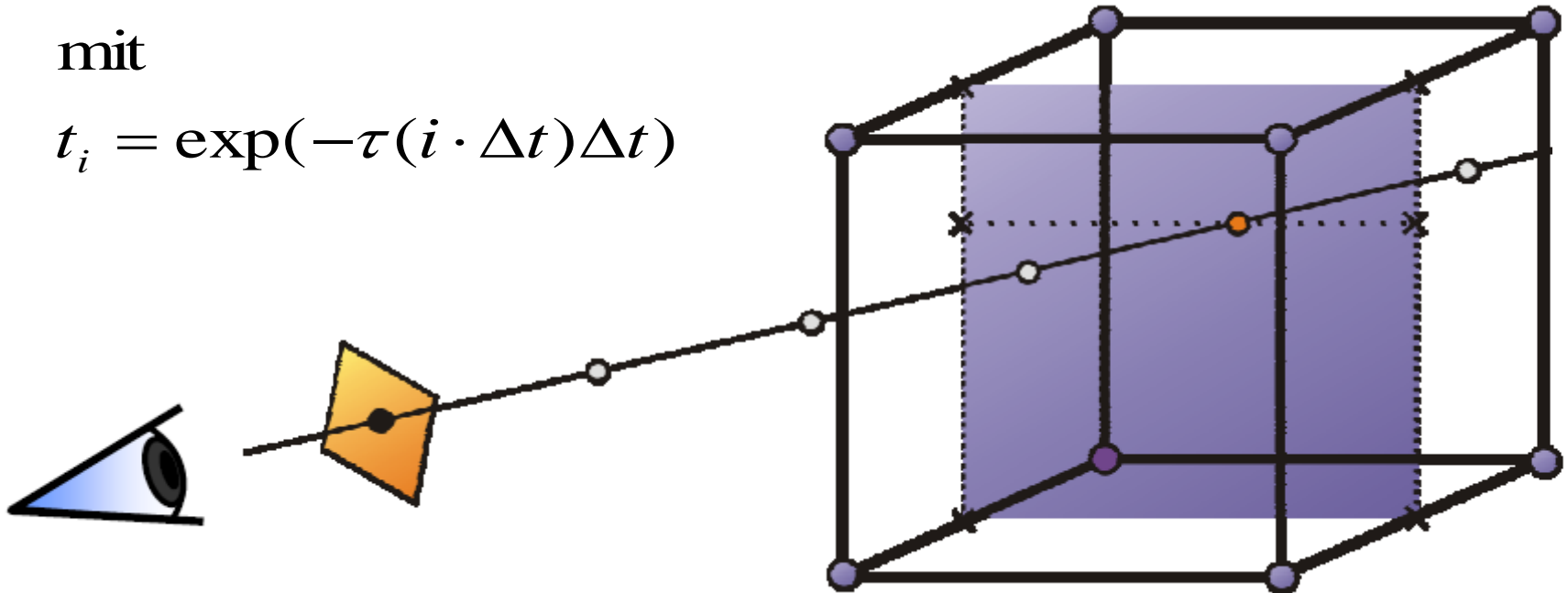
# Diskrete Auswertung der Volumen-Rendering-Gleichung

$$I(s) = I_{s_0} \cdot \exp\left(-\int_{s_0}^s \tau(t) dt\right) + \int_{s_0}^s Q(\tilde{s}) \cdot \exp\left(-\int_{\tilde{s}}^s \tau(t) dt\right) d\tilde{s}$$

$$I(s) = I_0 \prod_{k=0}^{n-1} t_k + \sum_{k=0}^{n-1} Q(k \cdot \Delta s) \cdot \Delta s \prod_{j=k+1}^{n-1} t_j$$

mit

$$t_i = \exp(-\tau(i \cdot \Delta t) \Delta t)$$



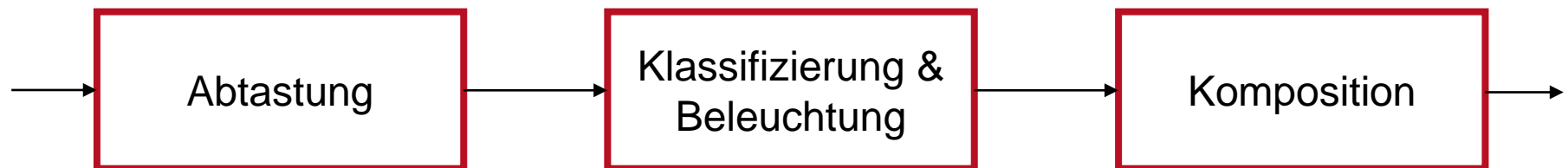


# Volumen-Rendering-Pipeline



- Drei Basisoperationen

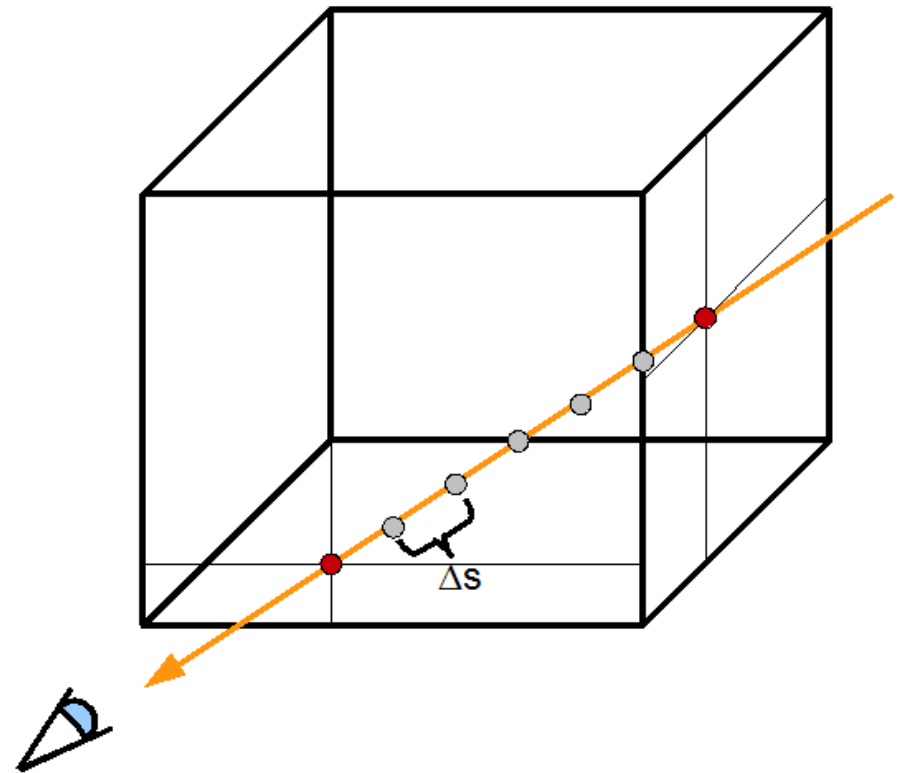
1. Abtastung (Sampling)  $\rightarrow k, \Delta s$
2. Klassifizierung und Beleuchtung  $\rightarrow Q(k), t_k$
3. Komposition  $\rightarrow \Sigma, \Pi$

$$I(s) = I_0 \prod_{k=0}^{n-1} t_k + \sum_{k=0}^{n-1} Q(k \cdot \Delta s) \cdot \Delta s \prod_{j=k+1}^{n-1} t_j$$



# 1. Abtastung

- Ansammlung von Voxelwerten an bestimmten Orten
- Die Position dieser Orte ist durch die Abtastdistanz  $\Delta s$  festgelegt
- Erinnerung: Shannons Sampling Theorem
  - ➔ Die Samplingdistanz sollte kleiner als die Hälfte der Rasterauflösung sein! 
- Abtastpositionen befinden sich (meistens) zwischen den Rasterpositionen
  - ➔ Wählen einer geeigneten Interpolationsmethode 



# 1. Abtastung - Aliasing



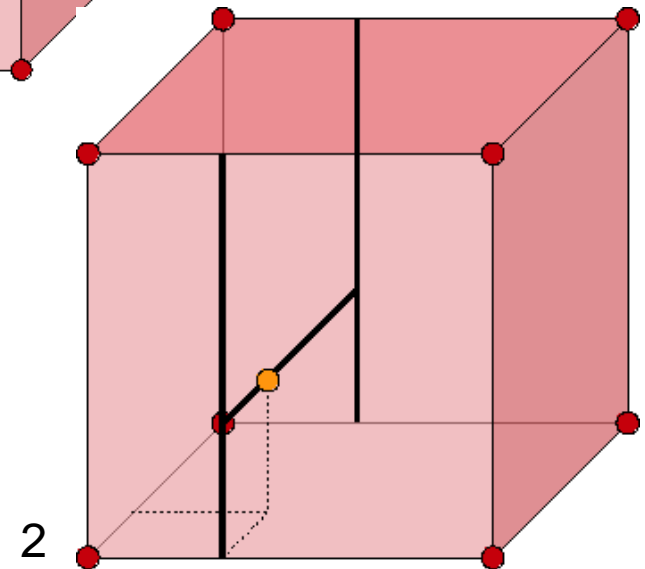
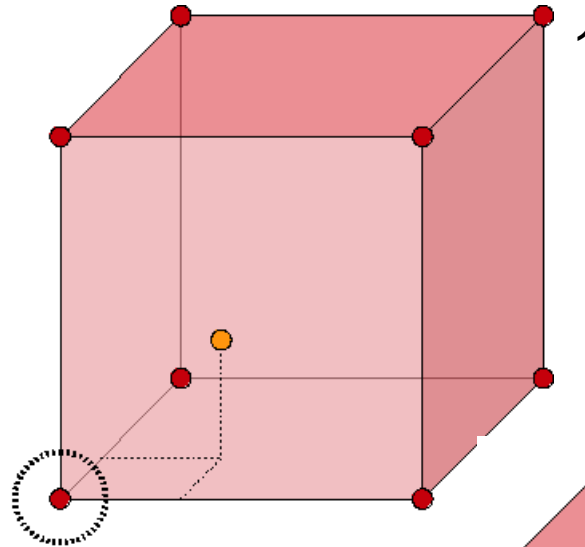
Unterhalb der Nyquist-Frequenz



Über der Nyquist-Frequenz

# 1. Abtastung - Interpolation

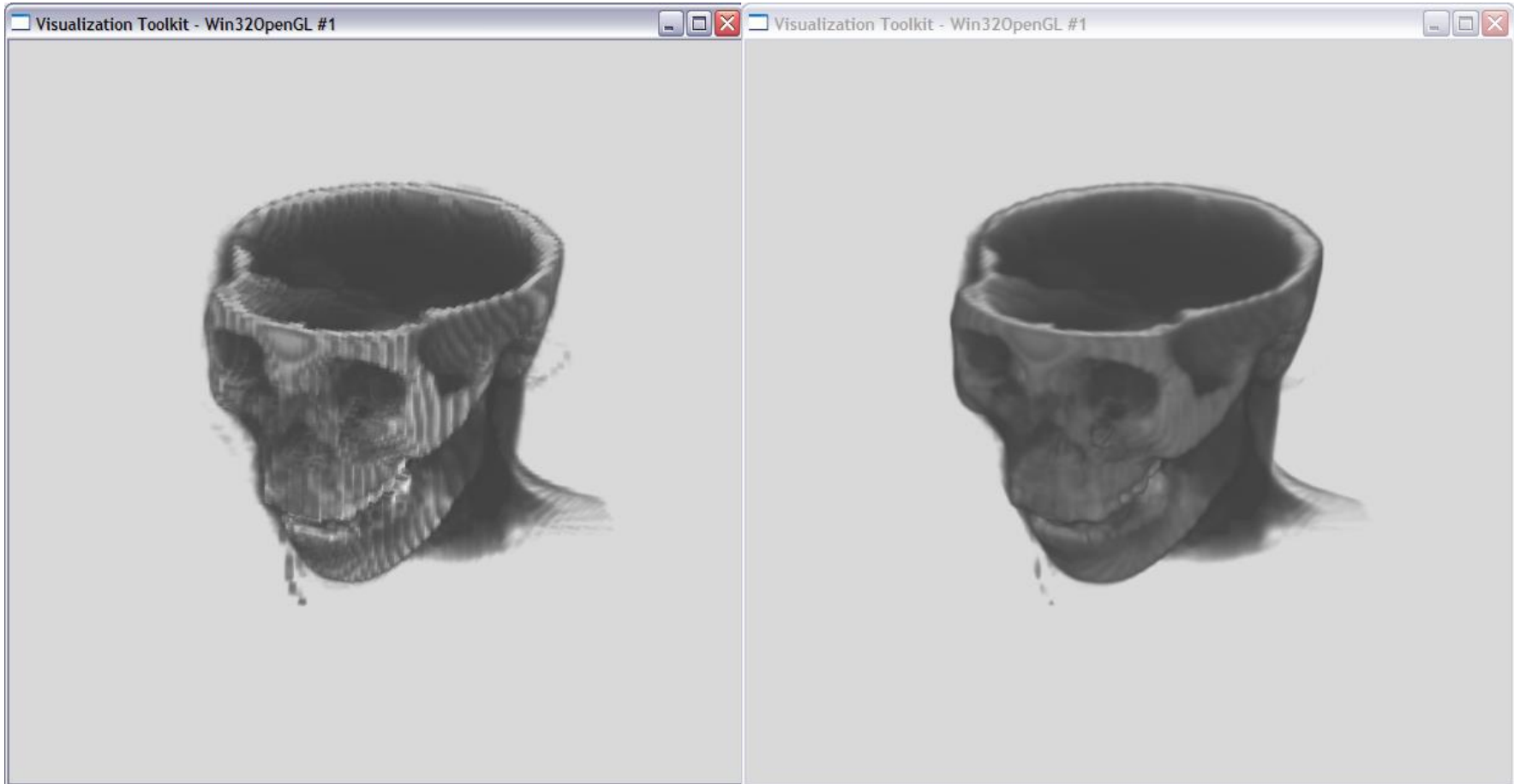
- Erhalten eines Wertes für eine Position zwischen Voxeln
- Unterschiedliche Interpolationsmodelle:
  1. Nearest Neighbor
  2. Trilinear
  3. B-Spline



# 1. Abtastung



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

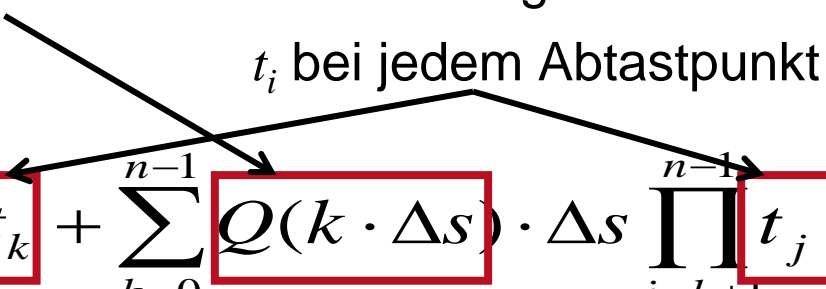


Nearest Neighbor-Interpolation

Trilineare Interpolation

## 2. Klassifikation und Beleuchtung

- Berechnen des Anteils  $Q_k$  jedes Abtastpunktes
- Bestimmung des Abschwächfaktors  $t_i$  bei jedem Abtastpunkt

$$I(s) = I_0 \prod_{k=0}^{n-1} t_k + \sum_{k=0}^{n-1} Q(k \cdot \Delta s) \cdot \Delta s \prod_{j=k+1}^{n-1} t_j$$


➡ Farbwert

➡ Transparenzwert

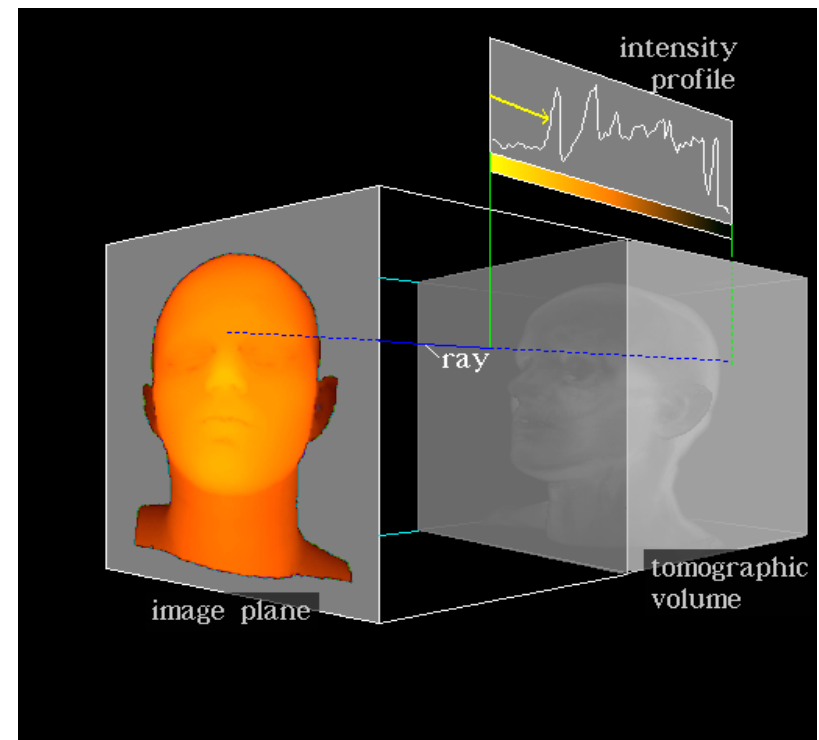
- Berechnen des beleuchteten Anteils
    - Volumenabtastungen werden als gerichtete Lichtquellen betrachtet
- ➔ Shading



### 3. Komposition

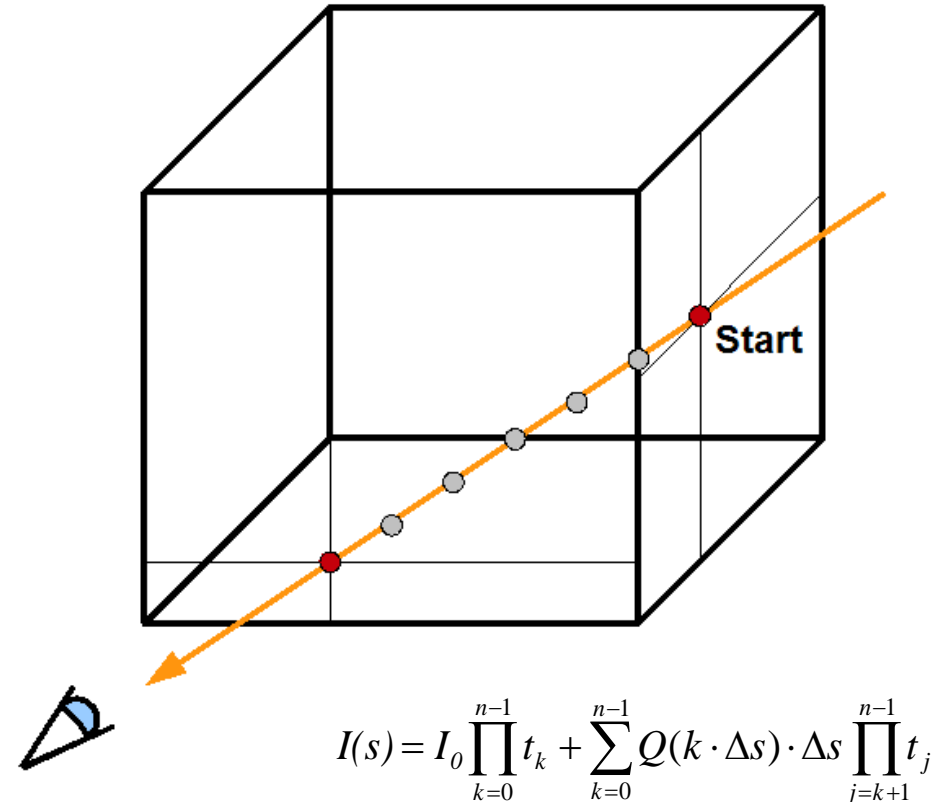
- Akkumulation der abgetasteten, klassifizierten und beleuchteten Werte
- Numerische Approximation der Volumen-Rendering-Gleichung

$$I(s) = I_0 \prod_{k=0}^{n-1} t_k + \sum_{k=0}^{n-1} Q(k \cdot \Delta s) \cdot \Delta s \prod_{j=k+1}^{n-1} t_j$$



## 3a. Back-to-Front-Komposition

- An der Abtastposition am Ende des Volumens mit dem Zusammensetzen beginnen
- In Richtung des Sichtpunktes gehen und dabei Anteile berechnen
- Anteil an Abtastposition  $k$ :
  - Vorige Anteile nach Transparenz  $t_k$  an der Abtastposition gewichten
  - + Farbe  $C_k$  an der aktuellen Abtastung

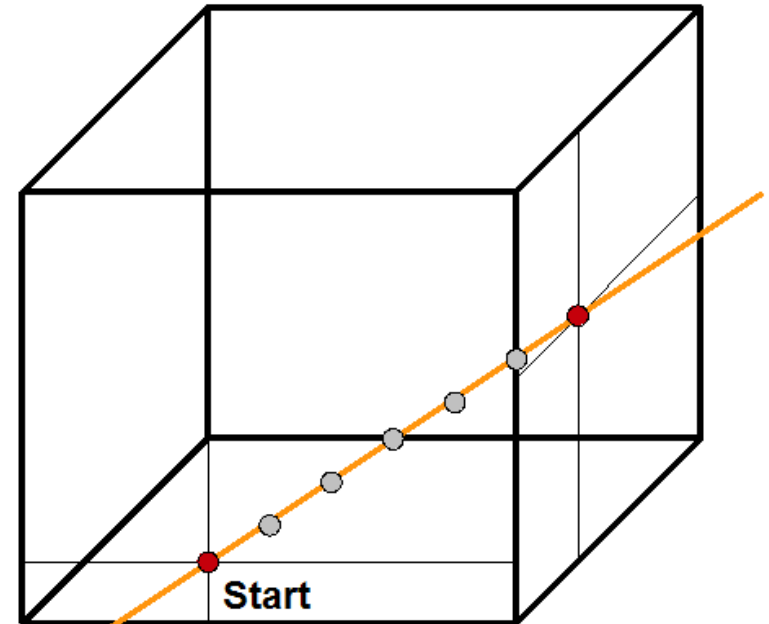


$$C_k = Q(k \cdot \Delta s) \cdot \Delta s$$

$$I_k = I_{k-1} \cdot t_k + C_k, \quad \forall k = 1, \dots, n-1$$

## 3b. Front-to-Back-Komposition

- An der Abtastposition am Anfang des Volumens mit dem Zusammensetzen beginnen
- In Richtung des Endes gehen und dabei Anteile berechnen
- Anteil an Abtastposition  $k$ :
  - Akkumulierte Farbe  $I_k$   
+ Farbe  $C_k$  gewichtet nach der akkumulierten Transparenz  $\tau_k$



$$I_{n-1} = C_{n-1}$$

$$\tau_{n-1} = t_{n-1}$$

$$I_{k-1} = I_k + C_k \cdot \tau_k, \quad \forall k = n-1, \dots, 1$$

$$\tau_{k-1} = t_k \cdot \tau_k, \quad \forall k = n-1, \dots, 1$$

$$I(s) = I_0 \prod_{k=0}^{n-1} t_k + \sum_{k=0}^{n-1} Q(k \cdot \Delta s) \cdot \Delta s \prod_{j=k+1}^{n-1} t_j$$

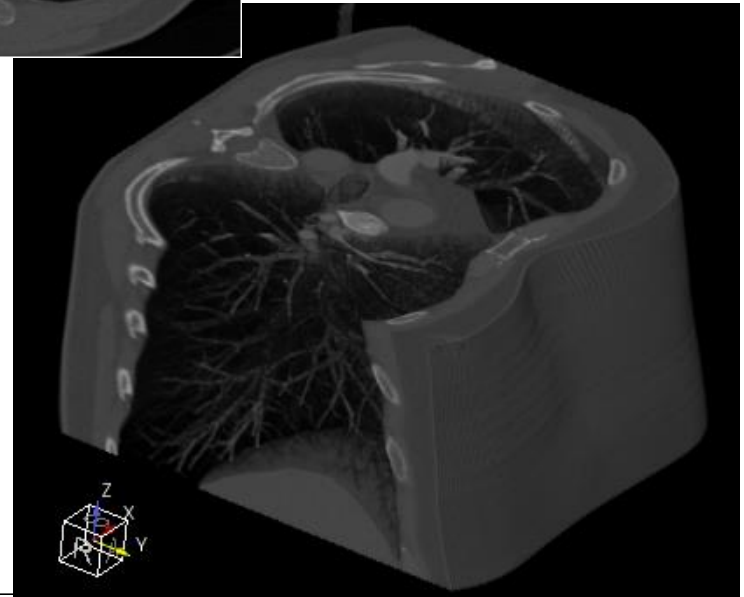
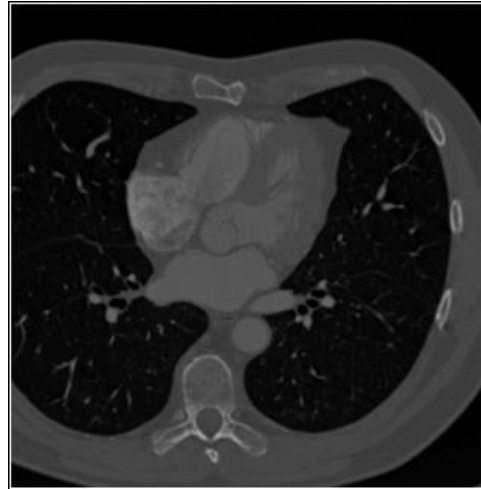
## 3b. Front-to-Back-Komposition

- Zwei akkumulierte Werte müssen berechnet werden:
  - Intensität
  - Transparenz
- Nebel:
  - Wird ab einer bestimmten Distanz undurchsichtig (opak)
  - Lichtquellen hinter diesem Punkt sind irrelevant



## 3b. Front-to-Back-Komposition

- Zusammensetzung stoppen, sobald die akkumulierte Transparenz zu klein wird → *Early Ray Termination*
  - Es müssen nicht alle Abtastpositionen entlang des Strahls betrachtet werden
  - Rendering-Geschwindigkeit kann verbessert werden
- Üblicherweise wird ein Transparenz-Threshold definiert



- Frage: Wie sollen gemessene und abgetastete Werte auf optische Eigenschaften abgebildet werden?

$$I(s) = I_0 \prod_{k=0}^{n-1} t_k + \sum_{k=0}^{n-1} Q(k \cdot \Delta s) \cdot \Delta s \prod_{j=k+1}^{n-1} t_j$$

## → Transferfunktionen

- Abgetastete Werte werden von einer Farbtransferfunktion klassifiziert  
→ Farbwert  $Q$
- Abgetastete Werte werden von einer Opazitätstransferfunktion klassifiziert  
→ Transparenzwert  $t$



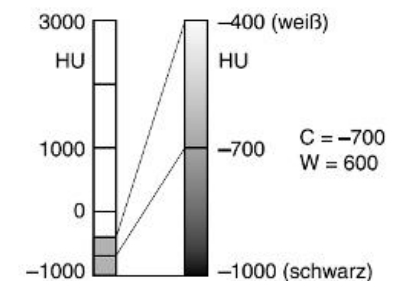
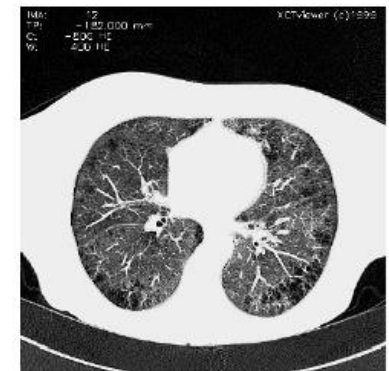
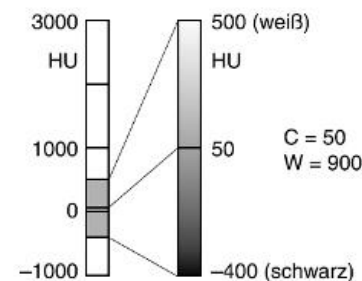
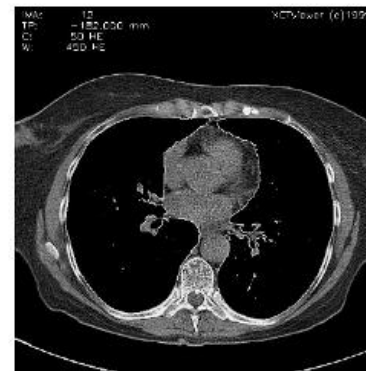
# Transferfunktionen

## Grundlagen

- Voxelwerte
  - ➔ Definitionsbereich der Funktion
- Optische Eigenschaften
  - ➔ Wertebereich der Funktion

$$tf_i : V \rightarrow O_i$$

- Eindimensionale Transferfunktion
- Erinnerung: Fensterung: CT-Daten werden oft von 4095 Grauwerten auf 8 Bit abgebildet





# Transferfunktionen

## Standardopazität und Grauwertabbildungen

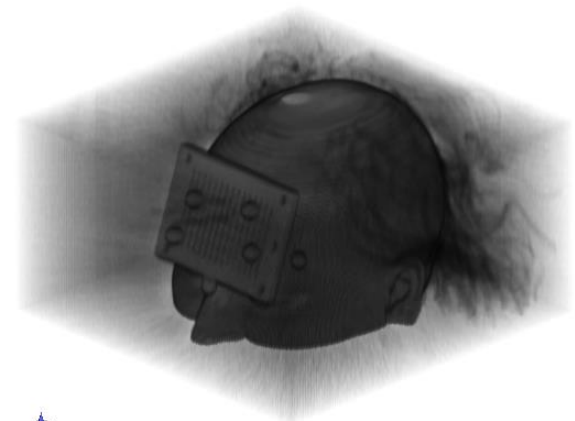
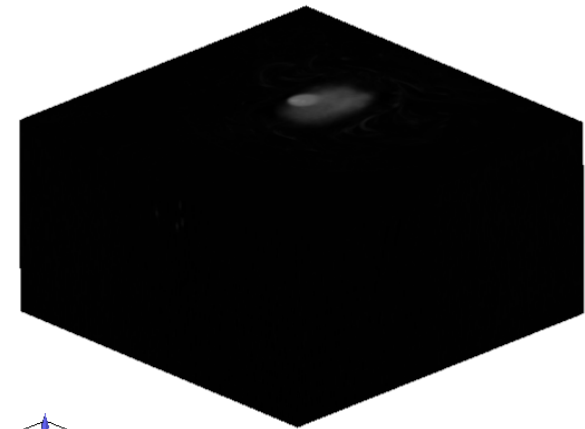
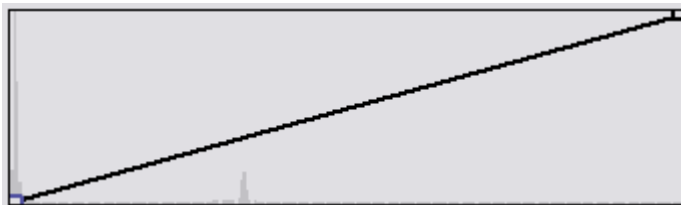


TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- Man betrachte nur Grauwerte
- Intensität wird gemäß den Abtastwerten zugewiesen
  - Der niedrigste Wert entspricht schwarz
  - Der höchste Wert entspricht weiß
  - Werte dazwischen werden linear interpoliert



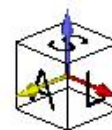
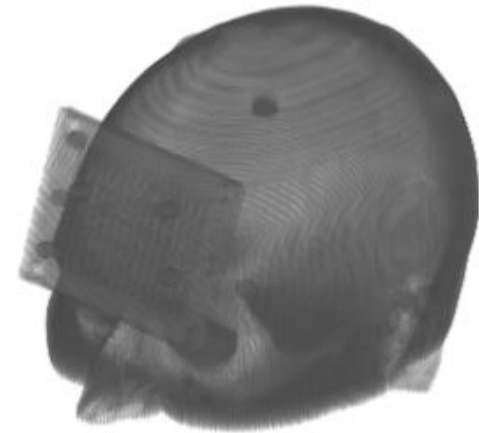
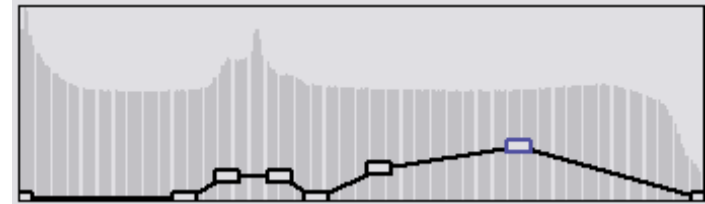
- Dunklen Werten wird geringe Opazität zugewiesen



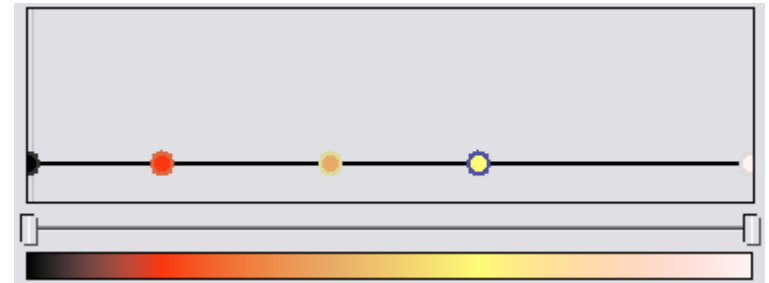
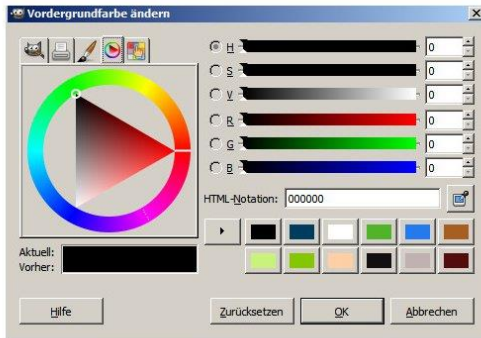
# Transferfunktionen Spezifikation



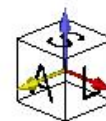
- Benutzung des Histogramms
- Referenzeinstellungen für Opazität und Farbe werden für eine begrenzte Anzahl von Werten im Definitionsbereich spezifiziert.
- Zwischen diesen Werten wird linear interpoliert.
- Höchstwert-Positionen werden oft gewählt, um Opazität zu erhöhen und Farbe zu ändern.



# Transferfunktionen Spezifikation

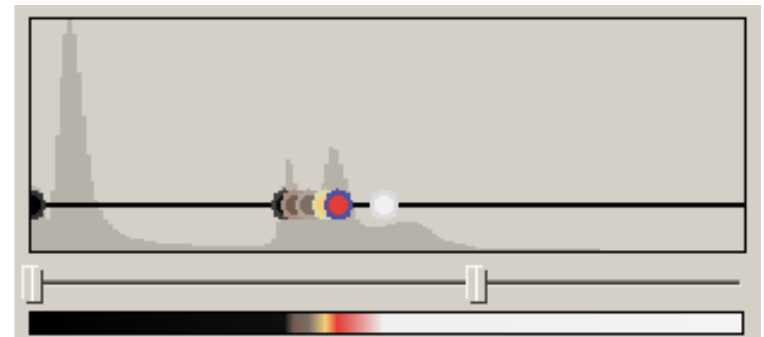
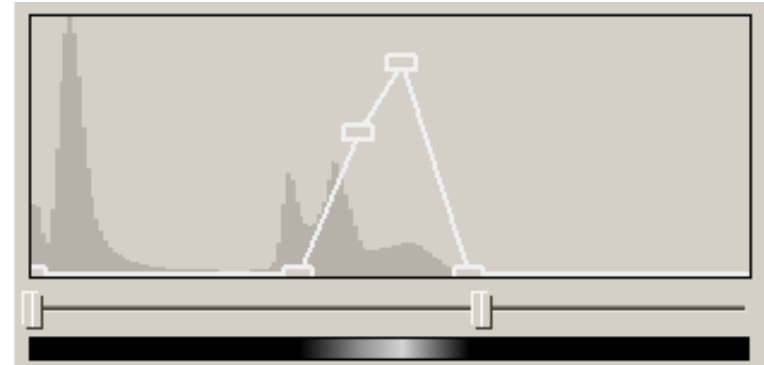
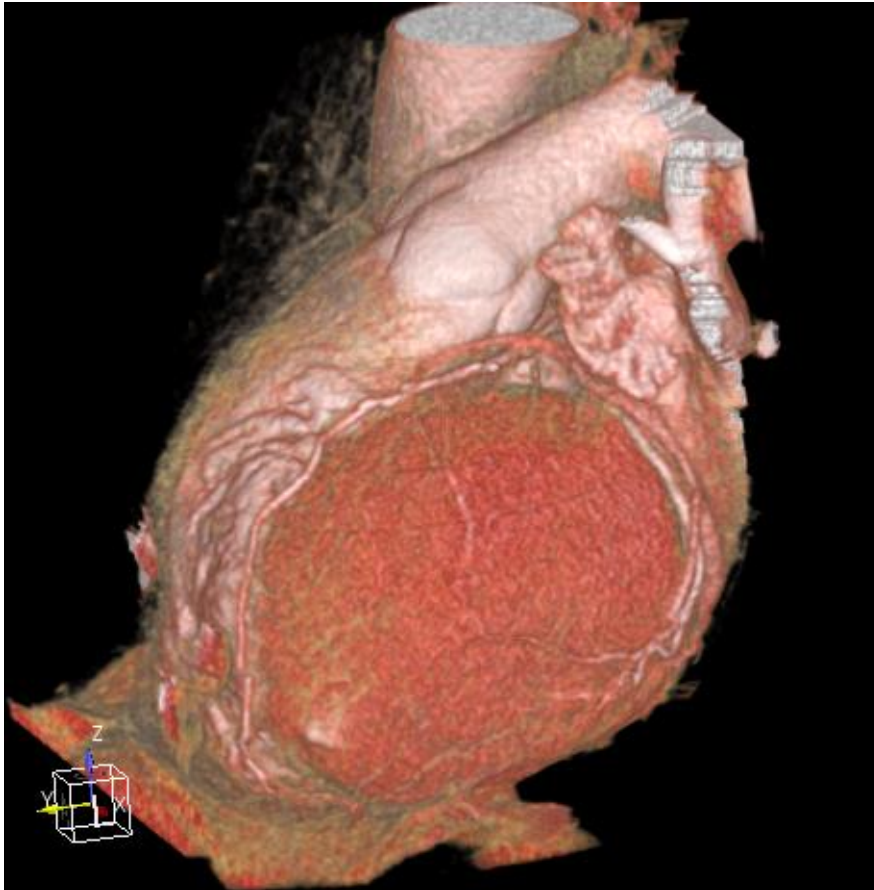


- Oft werden fotorealistische Renderings angepeilt
- Den verschiedenen Strukturen werden *echte* Farben zugewiesen
- Problem: Zwei Strukturen mit der selben Grauwertmenge haben verschiedene echte Farben

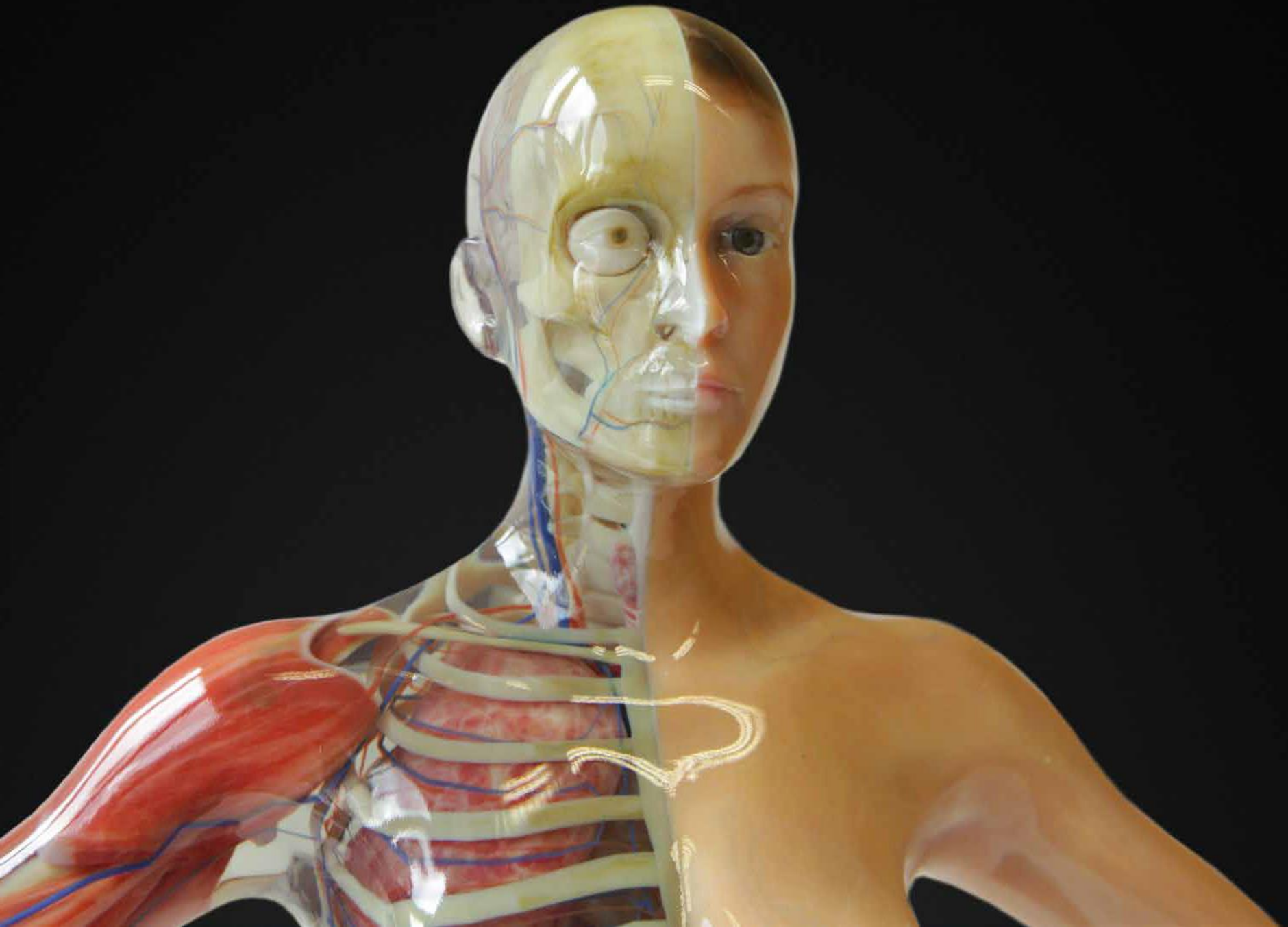


# Transferfunktionen

## Beispiel



▪ Shading

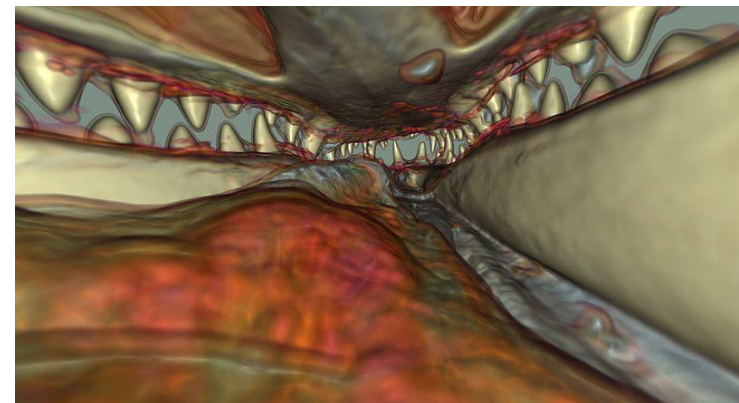
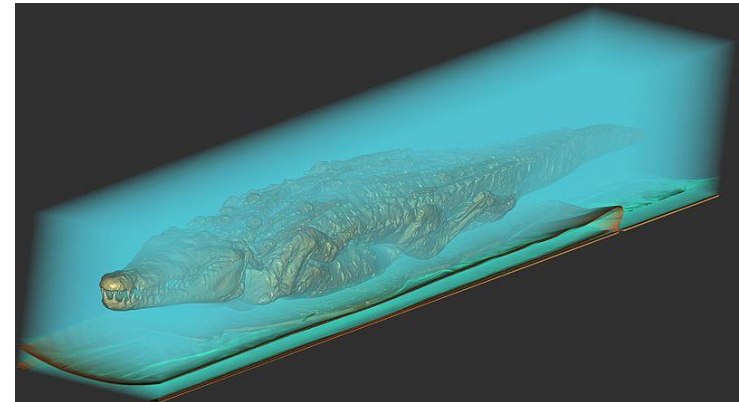
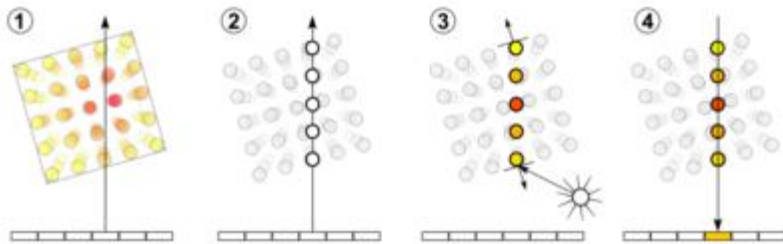




# Revisited: Volume ray casting

[http://en.wikipedia.org/wiki/Volume\\_ray\\_casting](http://en.wikipedia.org/wiki/Volume_ray_casting)

- The four basic steps of volume ray casting: (1) Ray Casting (2) Sampling (3) Shading (4) Compositing.



# Vielen Dank für die Aufmerksamkeit