

Visualisierung

Vorlesung im Wintersemester 2016/17

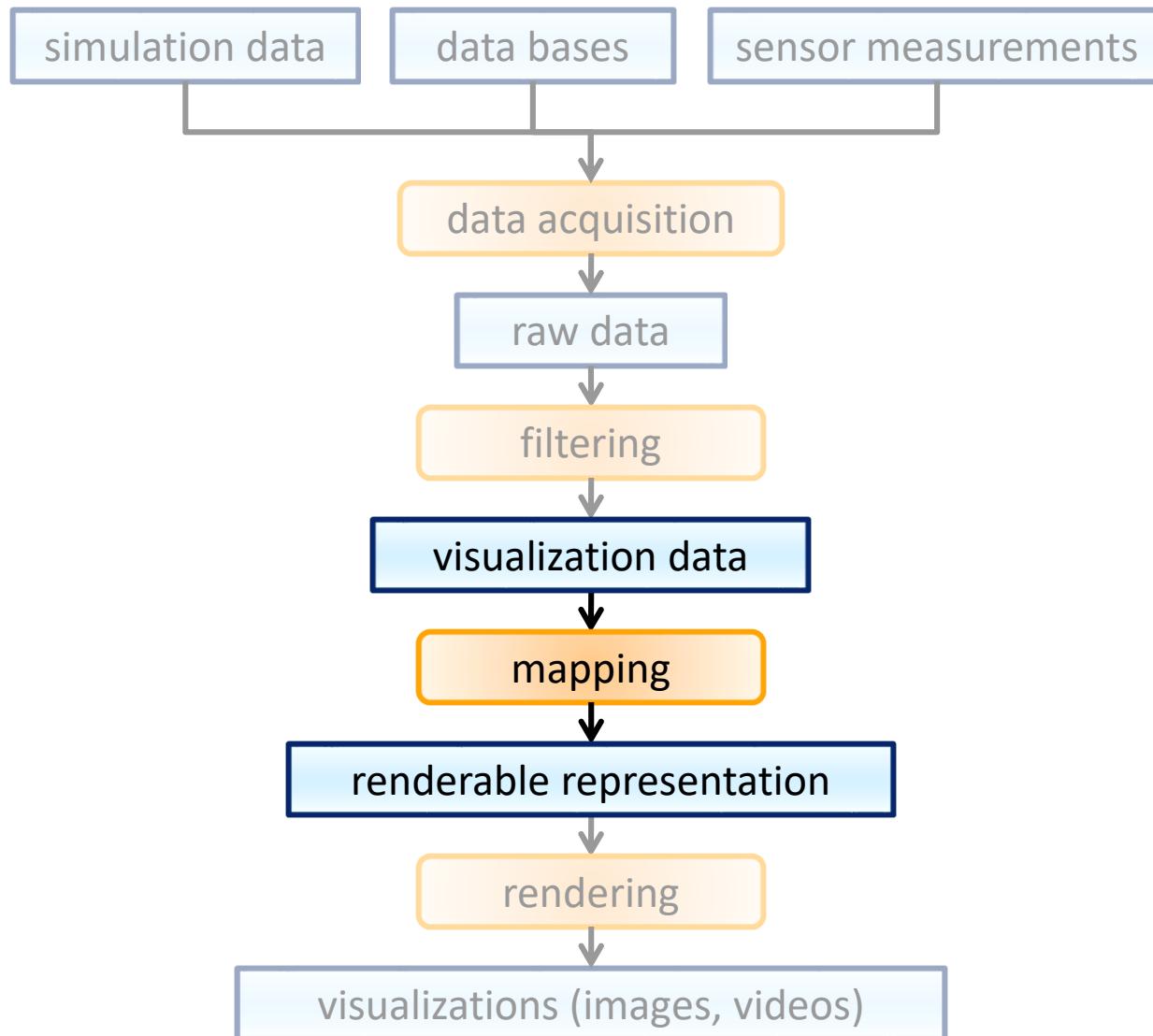
4) Skalardaten und Volumenvisualisierung

Prof. Dr.-Ing. Carsten Dachsbacher
Lehrstuhl für Computergrafik
Karlsruher Institut für Technologie



Mapping-Techniken

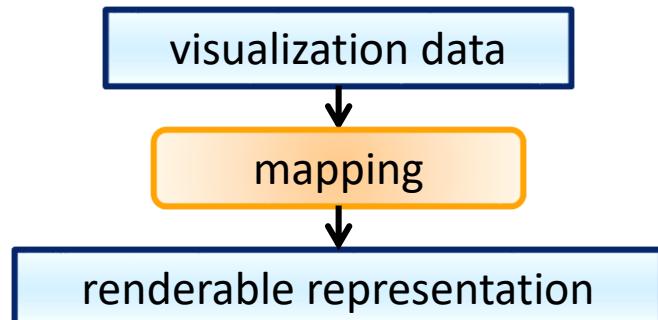
► wir befassen uns mit...



Visualisierungspipeline

Mapping: Kernkomponente des Visualisierungsprozess

- ▶ bildet aufbereitete Visualisierungsdaten auf grafische Primitive ab und legt deren Attribute (Farbe etc.) fest
 - ▶ grafische Primitive: Punkte, Linien, Flächen, Volumen, ...
 - ▶ Attribute: Farbe, Textur, Transparenz, ...
- ▶ Beispiele:
 - ▶ 2D-Skalarfeld → Höhenfeld, Farbdarstellung
 - ▶ 2D/3D-Skalarfeld → Isolinie/Isofläche
 - ▶ 3D-Skalarfeld → Volumenvisualisierung
 - ▶ Vektorfeld → Pfeile
 - ▶ Tensorfeld → Glyphen



2D-Skalarfelder

► Visualisierung von 2D-Skalarfeldern

► geg. eine Funktion $f: \mathbb{R}^2 \rightarrow \mathbb{R}$

► Visualisierung eines Attributs (= Skalarwert) mit Ortsbezug bzw. räumlicher Struktur

► Beispiele

► Farbmapping

► Höhenfelder

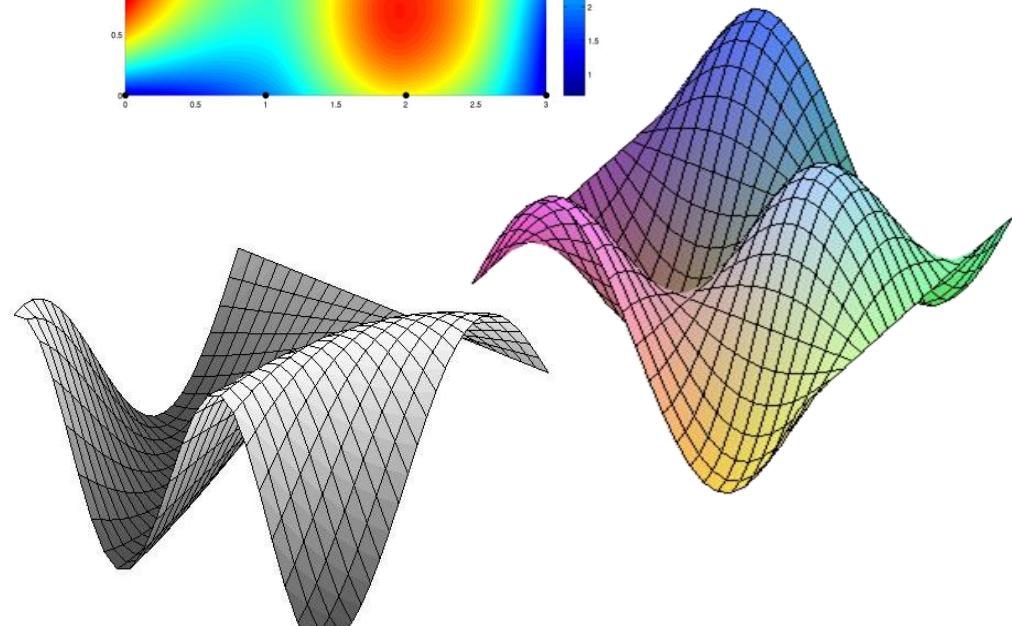
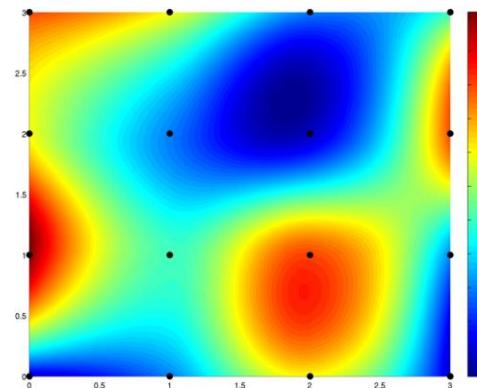
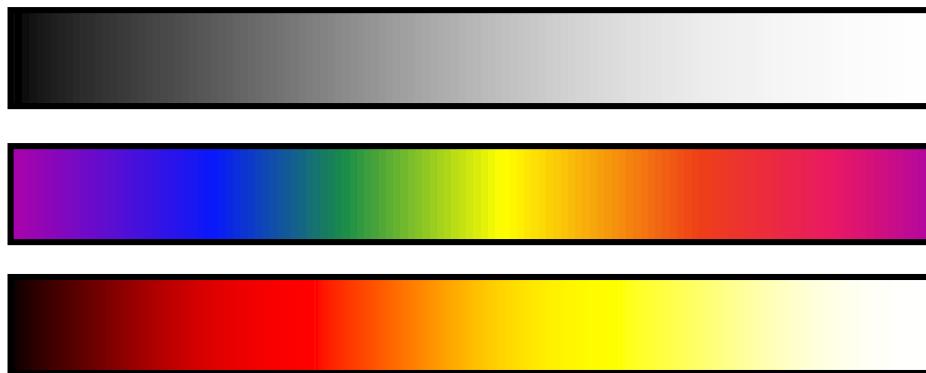


Abbildung auf Farbe

- Farbe ist eine retinale Variable (eigener „Wahrnehmungskanal“)
 - kann daher mit anderen Mapping-Techniken (z.B. räumliche Anordnung) kombiniert werden
- wichtige Mapping-Techniken basieren auf Farben und wir sind Farbkodierungen gewöhnt, z.B. warm = rot ↔ kalt = blau oder ok = grün ↔ nicht ok = rot, aber...
 - welche Farbbelegungen sind intuitiv oder zu vermeiden (Metaphern)?
 - welche Art von Daten sind überhaupt (effizient) farbkodierbar?
 - wie steht es mit Farbenblindem? Rot-Grün-Schwäche?



2D-Skalarfelder

► Visualisierung von 2D-Skalarfeldern

- ▶ geg. eine Funktion $f: \mathbb{R}^2 \rightarrow \mathbb{R}$
- ▶ Visualisierung eines Attributs (= Skalarwert) mit Ortsbezug bzw. räumlicher Struktur

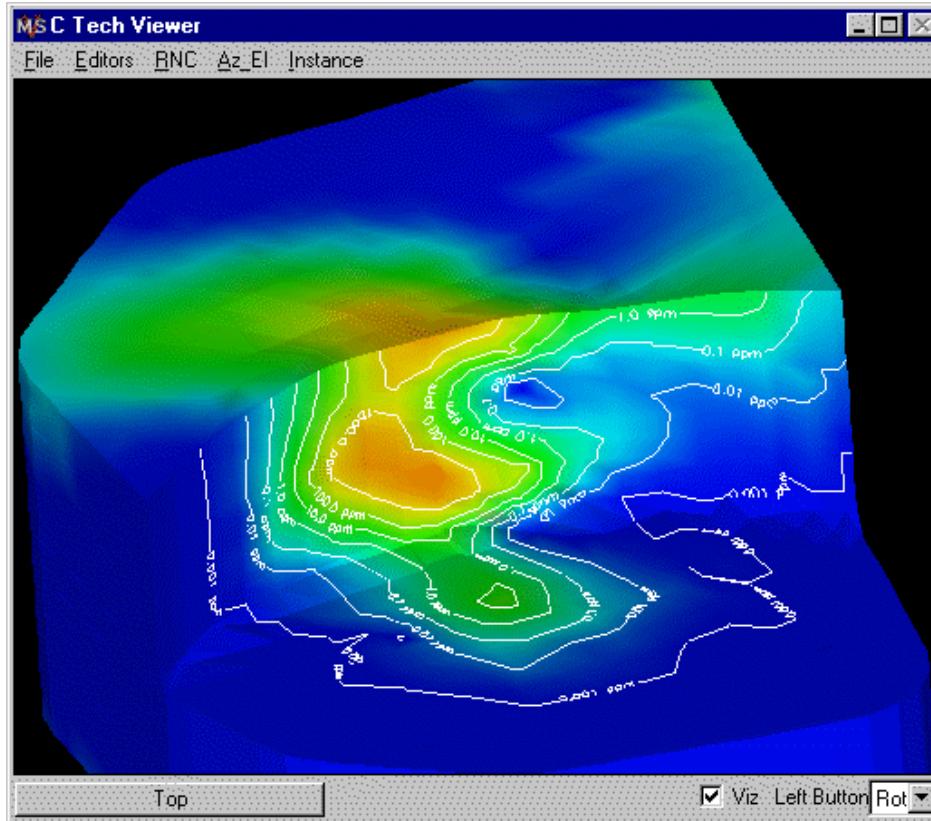
► Beispiele

- ▶ Farbmapping
- ▶ Höhenfelder
- ▶ Isolinie besteht aus den Punkten
 $\{(x, y) | f(x, y) = c\}$ für einen Skalarwert $c \in \mathbb{R}$



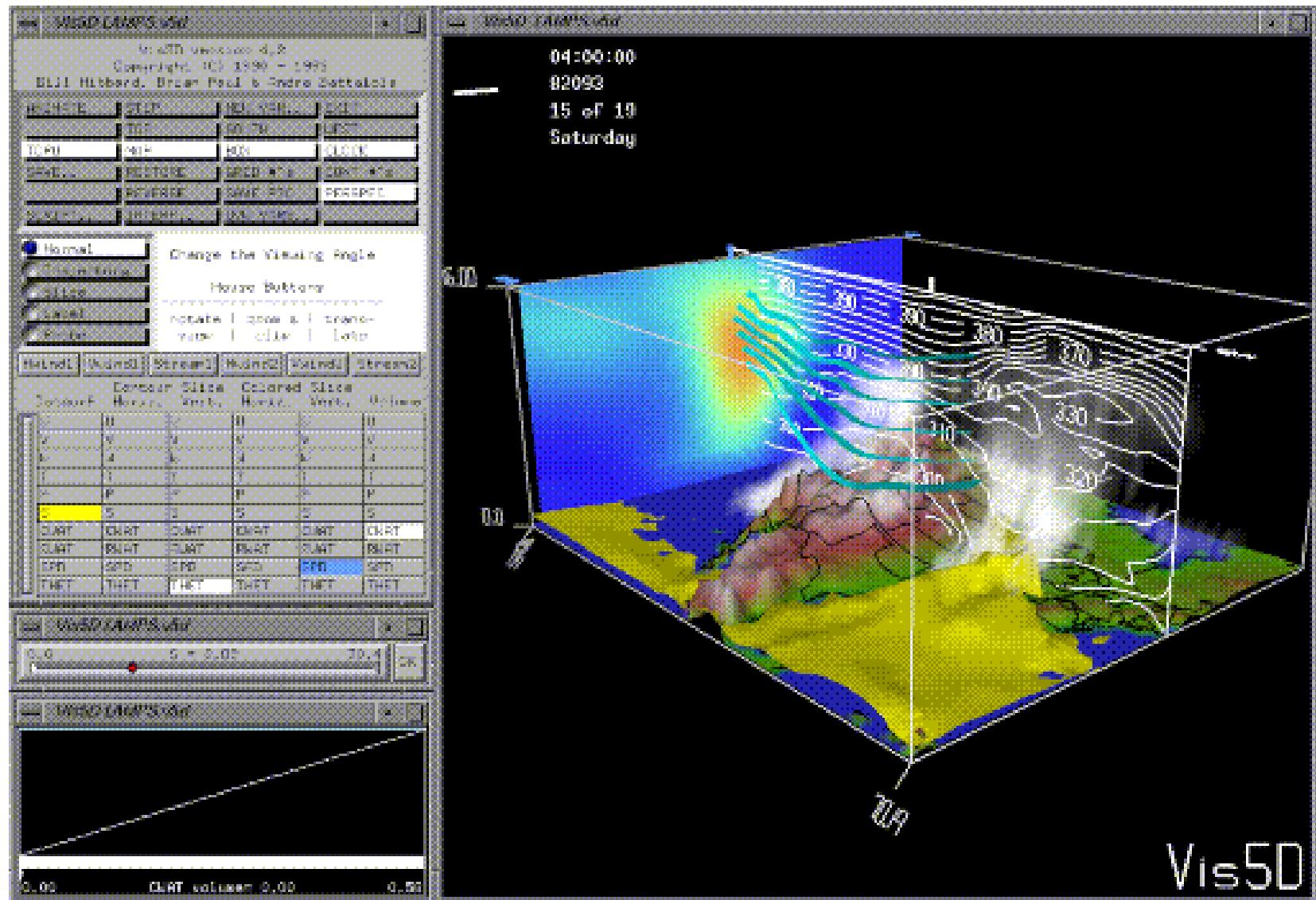
Isolinien

- Visualisierung von 2D-Skalarfeldern
 - geg. eine Funktion $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ (hier auf Schnittfläche) und ein Skalarwert $c \in \mathbb{R}$
 - die Isolinie (Isokontur, Konturlinie) besteht aus den Punkten $\{(x, y) | f(x, y) = c\}$



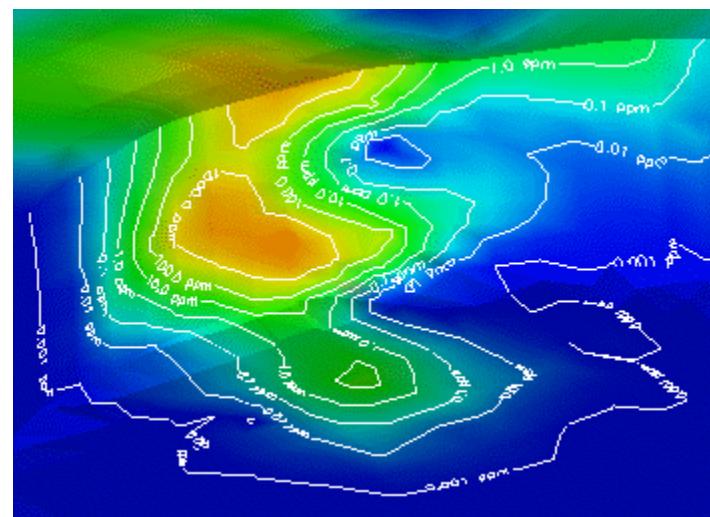
Isolinien

Beispiel: Wettermodelle



Isolinien

- Visualisierung von 2D-Skalarfeldern
 - geg. eine Funktion $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ und ein Skalarwert $c \in \mathbb{R}$
 - die Isolinie (Isokontur, Konturlinie) besteht aus den Punkten $\{(x, y) | f(x, y) = c\}$
- wenn f differenzierbar und $\text{grad } f \neq 0$, dann sind die Isolinien Kurven
 - wenn an einigen Stellen $\text{grad } f = 0$, dann können Isolinien einzelne Punkte oder Flächenstücke sein
- wichtig:
 - Isolinien sind geschlossene Kurven, oder
 - sie treten in den Beobachtungsraum ein und wieder aus (d.h. sie enden nie im Beobachtungsraum)



Isolinien

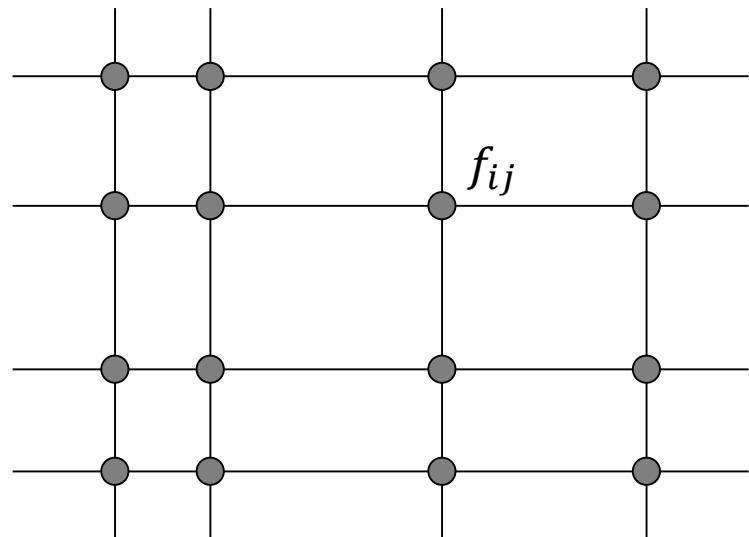
Pixel-by-Pixel Contouring

- einfachster Ansatz:
betrachte alle Pixel des Bildes und prüfe auf Gleichheit mit dem Isowert
- Eingabe
 - $f: (1, \dots, x_{max}) \times (1, \dots, y_{max}) \rightarrow \mathbb{R}$
 - Isowerte c_1, \dots, c_n und Isofarben c_1, \dots, c_n
- Pseudocode:

```
for all  $(x, y) \in (1, \dots, x_{max}) \times (1, \dots, y_{max})$  do
    for all  $k \in \{1, \dots, n\}$  do
        if  $|f(x, y) - c_k| < \epsilon$  then
            draw(  $x, y, c_k$  )
```
- Problem: die Isolinie kann natürlich verpasst werden, wenn der Gradient zu groß ist (trotz der ϵ -Toleranz)

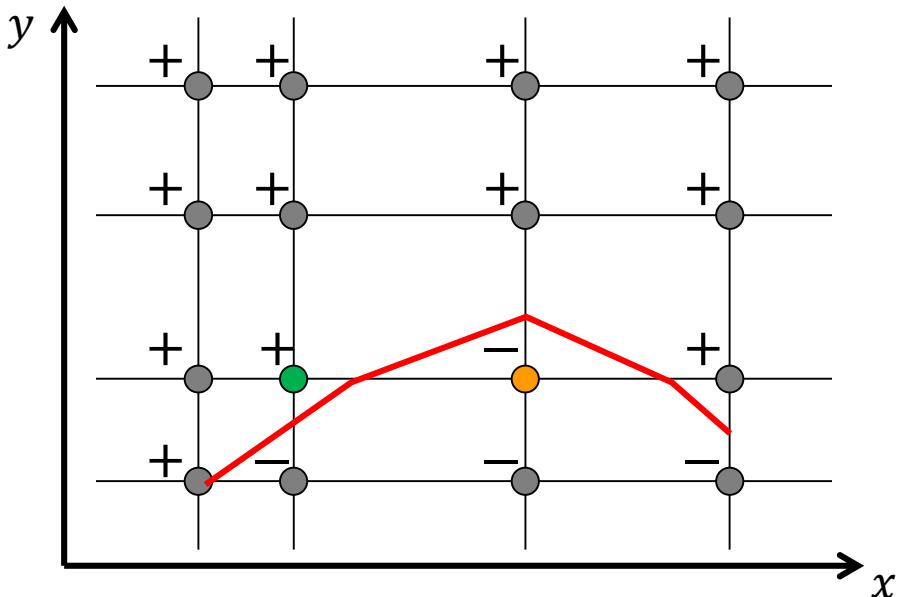
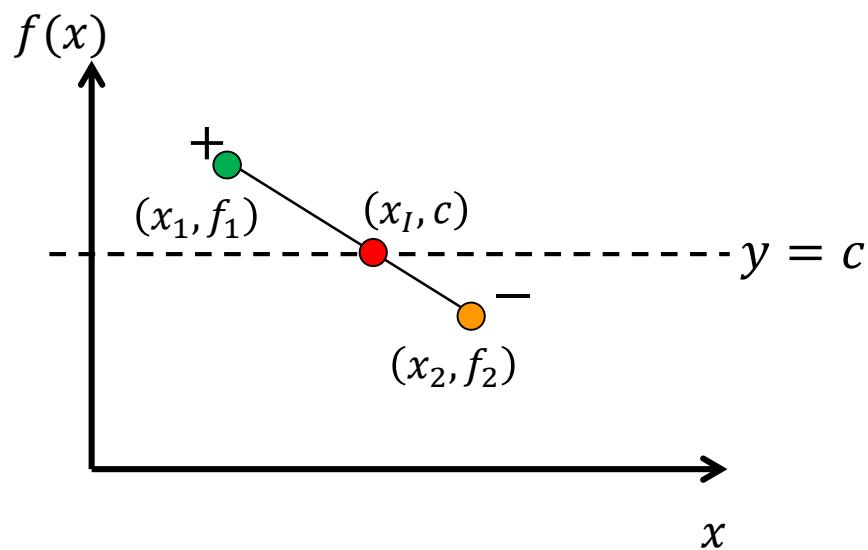
Marching Squares

- ▶ Repräsentation eines Skalarfelds als rektilineares Gitter (häufiger Fall)
 - ▶ Skalarwerte f_{ij} sind für jeden Knoten gegeben
 - ▶ Idee: betrachte die Interpolation innerhalb der Zellen (hier: bilinear), um Isolinien zu finden
- ▶ Vorteile von Marching Squares
 - ▶ Isolinien können nicht verpasst werden
 - ▶ Zellen können unabhängig voneinander (also parallel) betrachtet werden



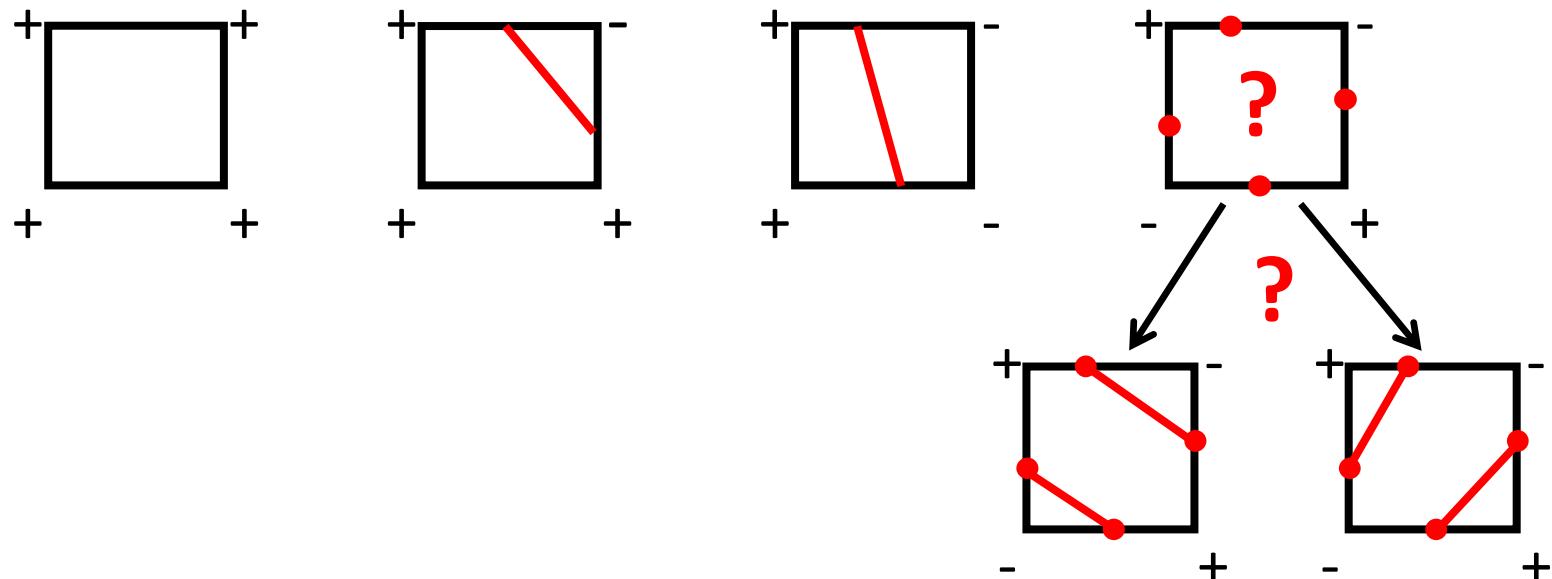
Marching Squares

- ▶ welche Zellen können überhaupt von der Isolinie geschnitten werden?
 - ▶ markiere alle Knoten mit + oder –, abh. davon ob $f_{ij} \geq c$ oder $f_{ij} < c$
- ▶ Zellen die das gleiche Vorzeichen an allen 4 Knoten haben werden von keiner Isolinie geschnitten
 - ▶ wir müssen nur Kanten mit unterschiedlichen Vorzeichen betrachten
 - ▶ und den Schnittpunkt mittels linearer Interpolation berechnen
(bilinare Interp. ist linear in den Horizontalen und Vertikalen)



Marching Squares: unterschiedliche Fälle

- ▶ es gibt nur 4 verschiedene Vorzeichen-Kombinationen
 - ▶ insgesamt 16 durch Symmetrien: Rotation, Reflexion, Wechsel $+\leftrightarrow -$
- ▶ Fallunterscheidung welche Kanten die Isolinie schneidet:
 - ▶ alle Knoten gleiche Klassifikation: Isolinie schneidet die Zelle nicht
 - ▶ genau ein Knoten anders klassifiziert: eindeutig
 - ▶ je zwei Knoten $+$ bzw. $-$ jeweils mit gemeinsamer Kante: eindeutig
 - ▶ im letzten Fall (rechts) gibt es zwei mögliche Linienpaare



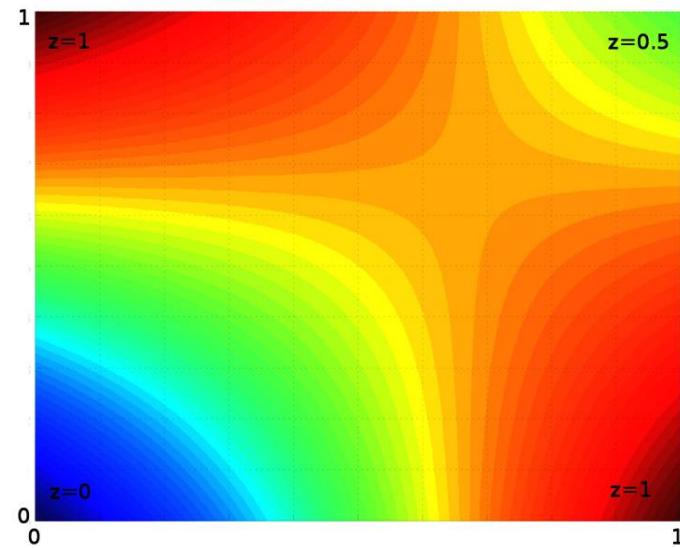
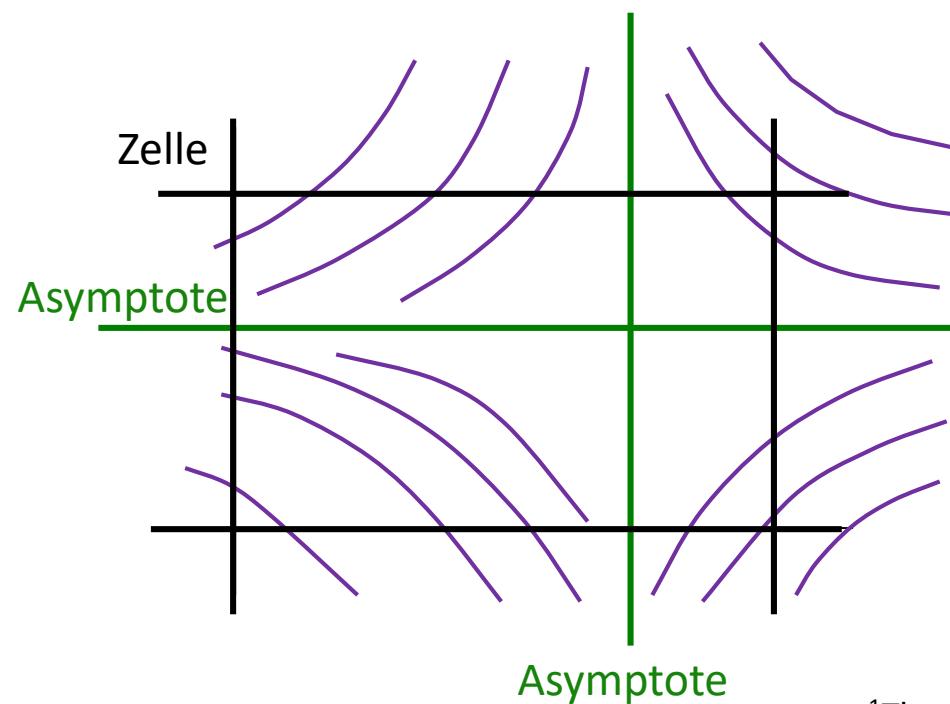
Isolinien

Marching Squares: Asymptotic Decider

- wir müssen uns ein Entscheidungskriterium überlegen...
- bilinearer Interpolant → Isolinien sind Hyperbeln¹

$$\{(x, y) | f(x, y) = c\}$$

$$f(x, y) = f_{i,j}(1-x)(1-y) + f_{i+1,j}x(1-y) + f_{i,j+1}(1-x)y + f_{i+1,j+1}xy$$



¹The asymptotic decider: resolving the ambiguity in marching cubes
Gregory M. Nielson, Bernd Hamann, VIS'91

Isolinien

Berechnung der Asymptoten

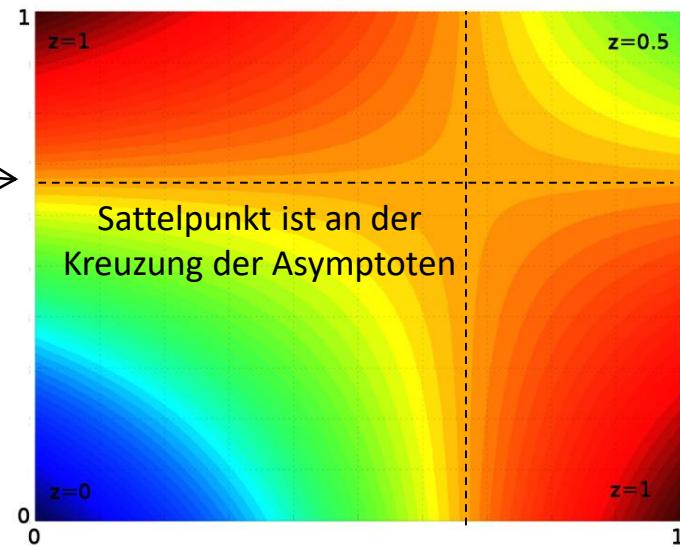
- ▶ $f(x,y) = f_{i,j}(1-x)(1-y) + f_{i+1,j}x(1-y) + f_{i,j+1}(1-x)y + f_{i+1,j+1}xy$
- ▶ $f(x,y) = axy + bx + cy + d$ mit

$$\begin{aligned} a &= f_{i,j} + f_{i+1,j+1} - f_{i+1,j} - f_{i,j+1} & b &= f_{i+1,j} - f_{i,j} \\ c &= f_{i,j+1} - f_{i,j} & d &= f_{i,j} \end{aligned}$$

- ▶ an den Asymptoten gilt:

$$\frac{df(x,y)}{dx} = ay + b \stackrel{!}{=} 0 \quad \text{keine Änderung in } x\text{-Richtung}$$

$$\frac{df(x,y)}{dy} = ax + c \stackrel{!}{=} 0$$



- ▶ Asymptoten:

$$y = -\frac{b}{a} = \frac{f_{i,j} - f_{i+1,j}}{f_{i,j} + f_{i+1,j+1} - f_{i+1,j} - f_{i,j+1}}$$

$$x = \dots$$

Isolinien

Berechnung der Asymptoten

- ▶ $f(x,y) = f_{i,j}(1-x)(1-y) + f_{i+1,j}x(1-y) + f_{i,j+1}(1-x)y + f_{i+1,j+1}xy$
- ▶ $f(x,y) = axy + bx + cy + d$ mit

$$\begin{aligned} a &= f_{i,j} + f_{i+1,j+1} - f_{i+1,j} - f_{i,j+1} & b &= f_{i+1,j} - f_{i,j} \\ c &= f_{i,j+1} - f_{i,j} & d &= f_{i,j} \end{aligned}$$

- ▶ an den Asymptoten gilt:

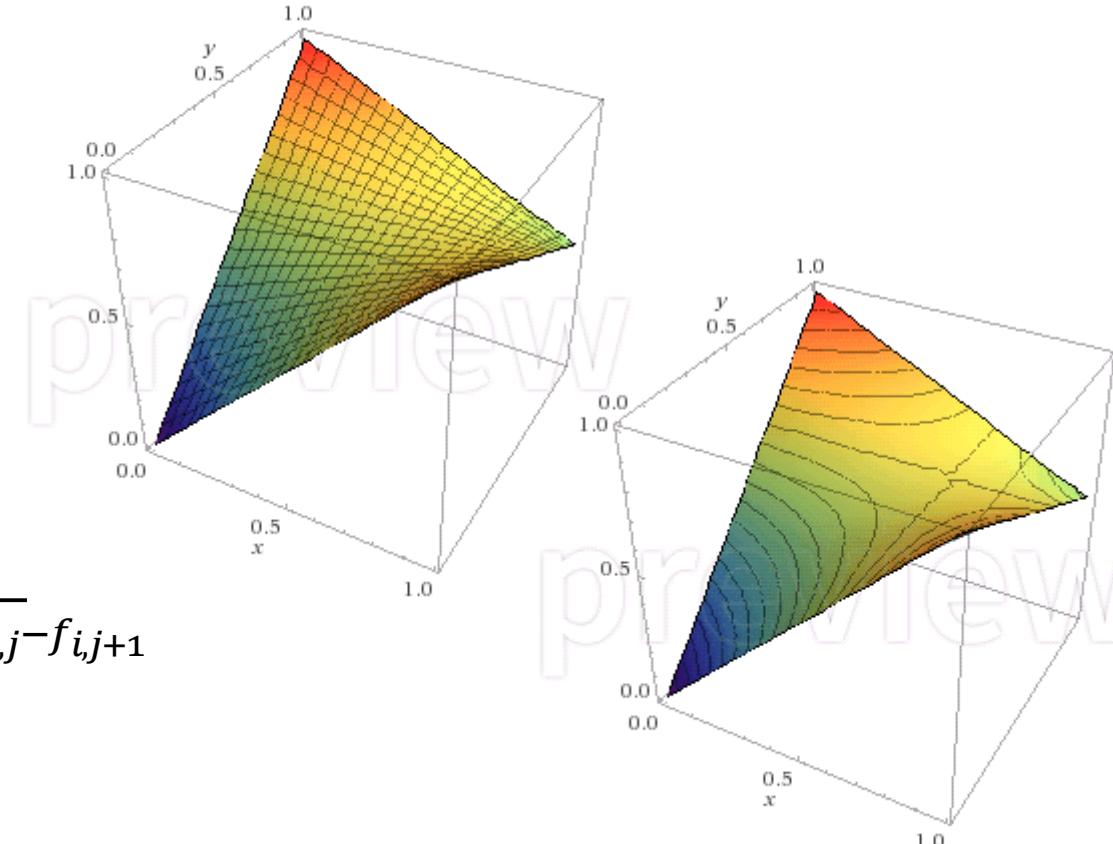
$$\frac{df(x,y)}{dx} = ay + b = 0$$

$$\frac{df(x,y)}{dy} = ax + c = 0$$

- ▶ Asymptoten:

$$y = -\frac{b}{a} = \frac{f_{i,j}-f_{i+1,j}}{f_{i,j}+f_{i+1,j+1}-f_{i+1,j}-f_{i,j+1}}$$

$$x = \dots$$

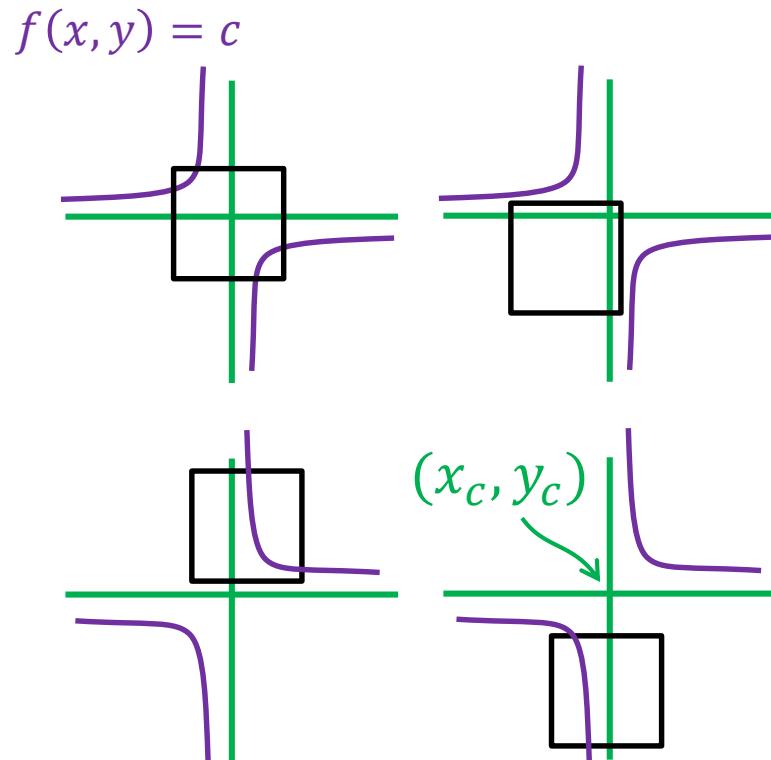


Isolinien

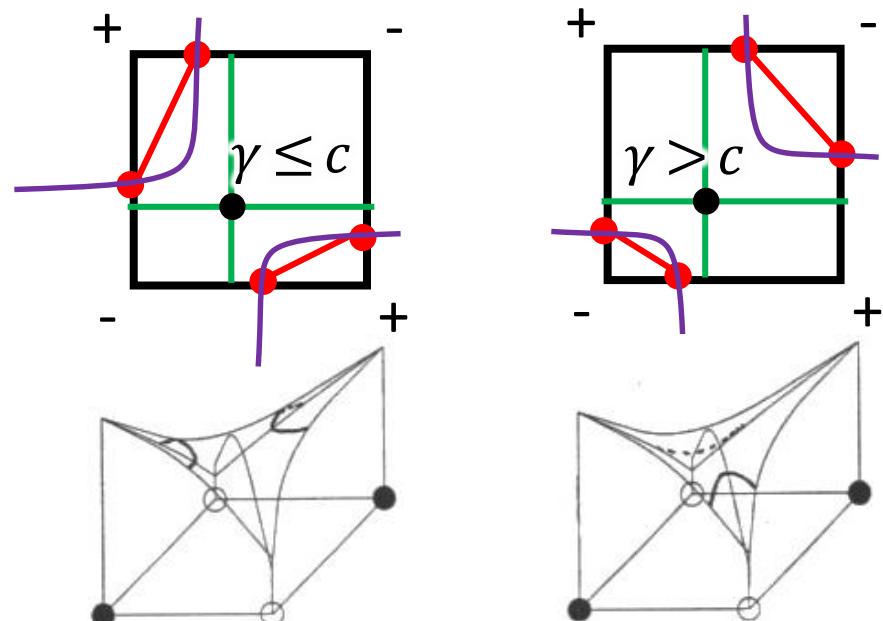
Asymptotic Decider

- ▶ mehrdeutige Fälle: wenn beide Äste der Hyperbel die Zelle schneiden
- ▶ Schritt 1: berechne den Sattelpunkt (x_c, y_c)

$$\blacktriangleright x_c = \frac{f_{i,j} - f_{i,j+1}}{f_{i,j} + f_{i+1,j+1} - f_{i,j+1} - f_{i+1,j}} \text{ und } y_c = \frac{f_{i,j} - f_{i+1,j}}{f_{i,j} + f_{i+1,j+1} - f_{i,j+1} - f_{i+1,j}}$$



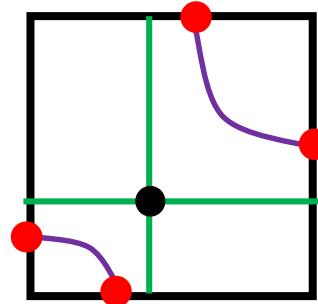
- ▶ Schritt 2: berechne $\gamma = f(x_c, y_c)$
- ▶ Schritt 3: entscheide



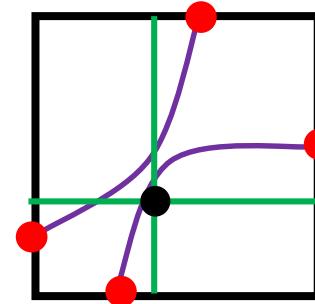
Isolinien

Asymptotic Decider

- es stellt sich heraus, dass die Lösung im 2D-Fall viel einfacher ist:
man betrachtet einfach die Reihenfolge (entlang der x - oder y -Richtung)
der Schnittpunkte mit den Isolinien



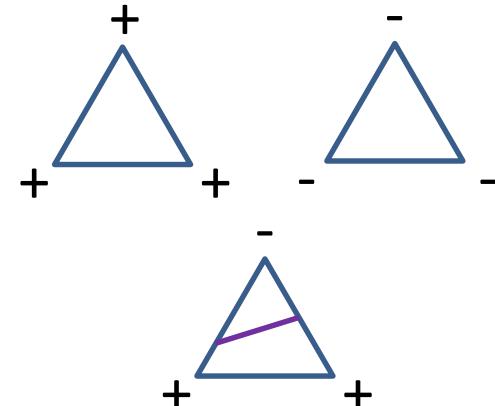
richtig



falsch / nicht möglich!
(kein asymptotisches Verhalten)

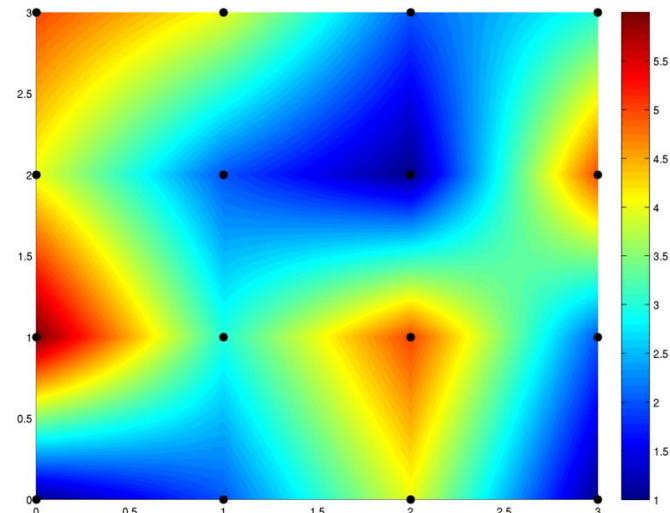
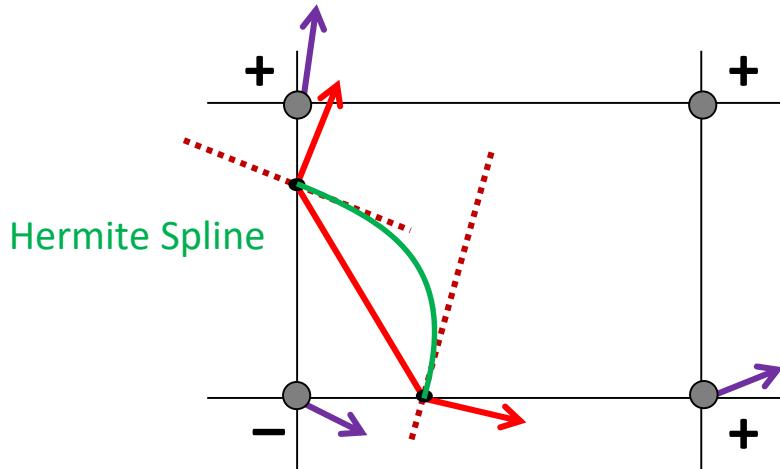
Isolinien in Triangulationen (Dreiecksnetzen)

- zwei eindeutige Fälle ohne Isoline
- ein eindeutiger Fall mit Isoline
(linearer Interpolant \rightarrow gerade Isolinie in der Zelle)



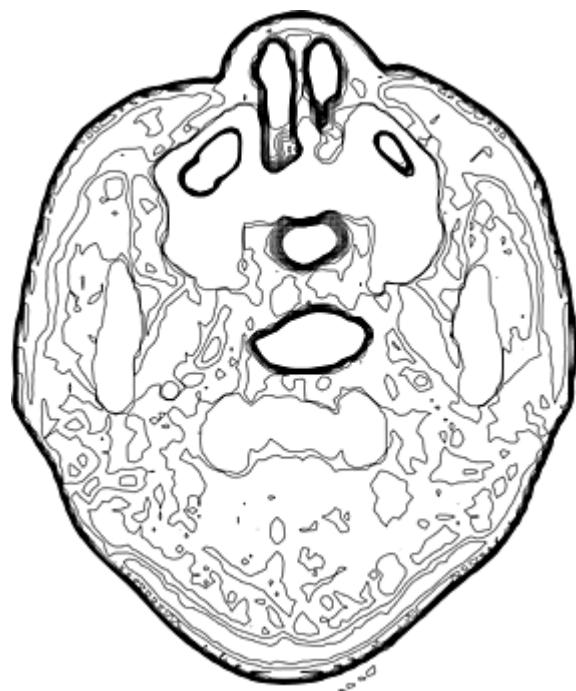
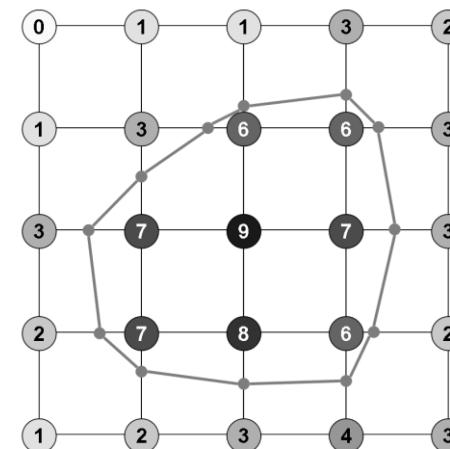
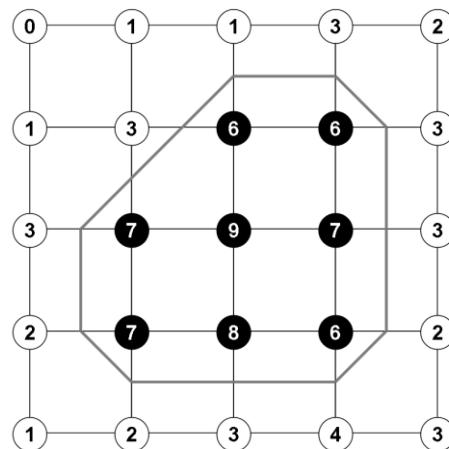
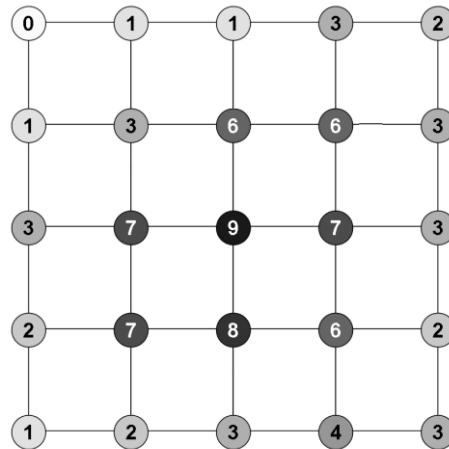
Glätten von Isolinien

- ▶ berechne die Gradienten an den Knoten (z.B. zentrale Differenzen)
- ▶ interpoliere die Gradienten für die Start-/Endpunkte der Isoliniensegmente linear entlang der Kanten
- ▶ Gradienten stehen senkrecht auf Isolinien
- ▶ zeichne Spline, Bézier-Kurve, ...



Beachte: Unstetigkeiten bei der bilinearen Interpolation an Zellengrenzen

Isolinien



Bemerkungen

- ▶ oft Darstellung mehrerer Isolinien mit unterschiedlichen Parametern:
Linienart (gestrichelt, gepunktet, ...), Farbe, ...
- ▶ Vorteile
 - ▶ auch Darstellung ohne Farbe möglich (ein visueller Kanal „gespart“)
 - ▶ Annotationen möglich
 - ▶ stellt implizit den Gradienten des Skalarfelds dar (Abstand von mehreren Isolinien)
 - ▶ erweiterbar auf 2D-Flächen in \mathbb{R}^3 (Triangulierung einer Oberfläche)
 - ▶ erweiterbar auf verschiedene Gitterstrukturen (Rechteck, Dreieck) und 3D-Gitter (Tetraeder, rektilineare Gitter)
- ▶ Nachteile
 - ▶ Minimum/Maximum nicht direkt erkennbar
 - ▶ Darstellung hängt stark von der Wahl des Isowerts ab

Volumenvisualisierung

- VolVis = Visualisierung von Volumendaten
 - ▶ Volumendaten sind Skalardaten in 3D, also $h: \mathbb{R}^3 \rightarrow \mathbb{R}$
 - ▶ bei der Darstellung: eine Abbildung von 3D Daten \rightarrow 2D Bild
 - ▶ durch Projektion (z.B. Maximum Intensity), Slicing, (fotorealistisches) Volumenrendering

- Daten stammen z.B. aus
 - ▶ Medizin: Computertomographie, Magnetresonanztomographie, ...
 - ▶ Materialüberprüfung (Industrie-CT)
 - ▶ Simulationen (Finite-Elemente Methoden, Strömungs-/Materialsimulation) etc.

- Ziele
 - ▶ den 3D Datensatz verstehen, interpretieren, analysieren
 - ▶ Strukturen von besonderem Interesse untersuchen und Kontext herstellen

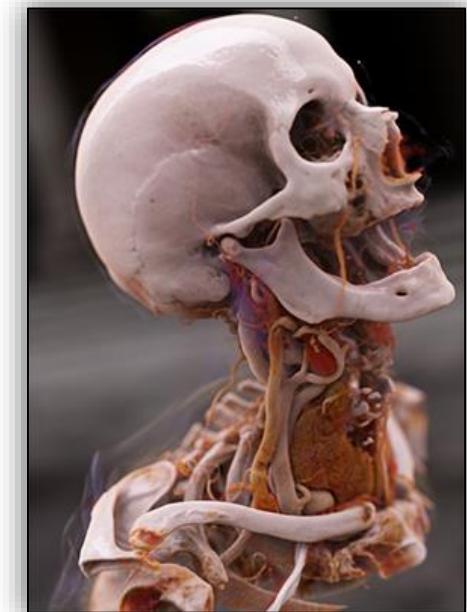


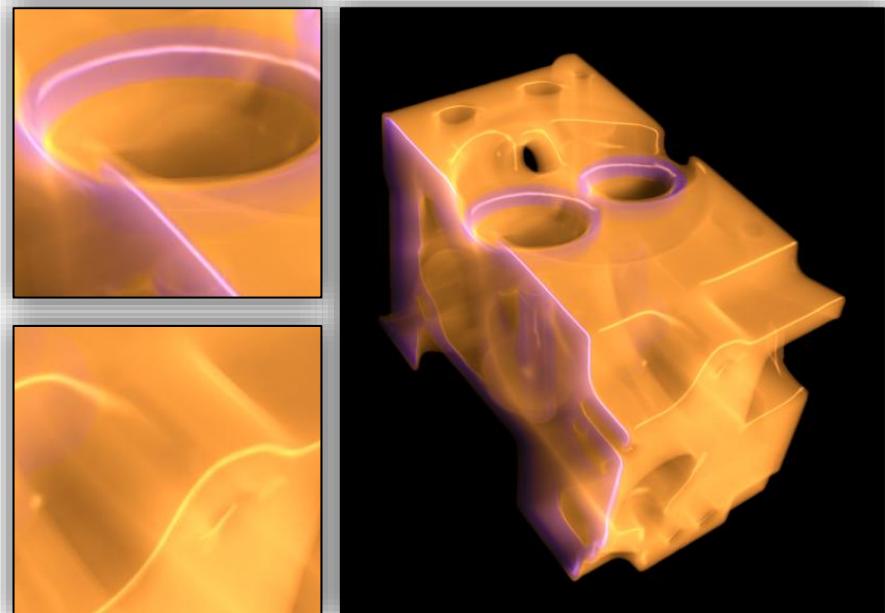
Bild: Siemens Cinematic Rendering

Volumenvisualisierung

- ▶ Einführung zu Volumenvisualisierung
 - ▶ Volumendaten
 - ▶ einfache Methoden (Slicing, ...)
 - ▶ Oberflächen- vs. Volumenrendering
 - ▶ Überblick über die Techniken
- ▶ Direkte Volumenvisualisierung
 - ▶ Transferfunktionen
 - ▶ ...
- ▶ Indirekte Volumen Visualisierung (z.B. über Isoflächen)



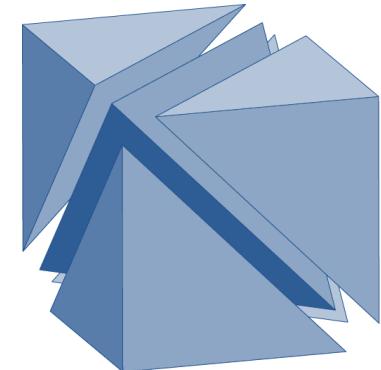
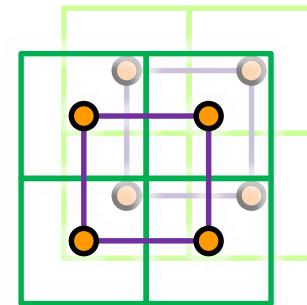
Bild: Siemens Cinematic Rendering



3D Volumendaten

Daten liegen für die Visualisierung i.d.R. vor als...

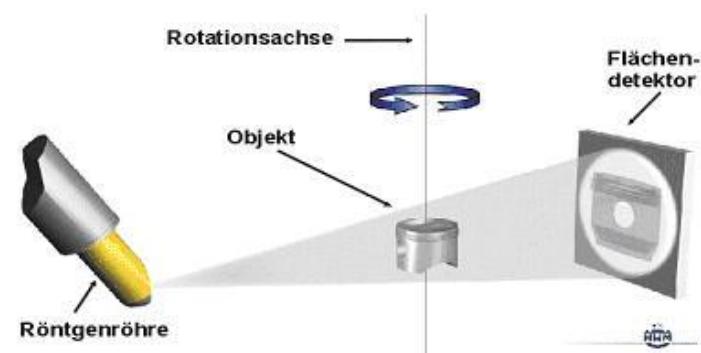
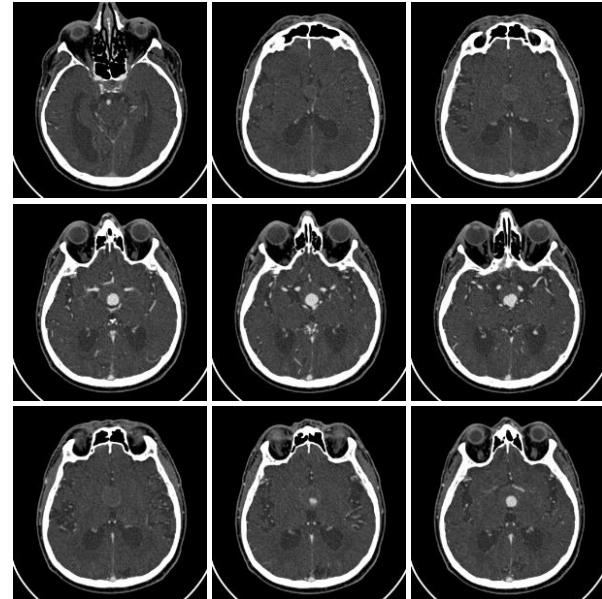
- ▶ ... kartesisches oder zumindest uniformes Gitter
 - ▶ bei CT/MR ist oft $dx = dy < dz$, also z.B. 140 Slices (in z-Richtung) à 512^2 Werte (als Pixel in einem Schichtbild)
 - ▶ Iso-Stack-Calculation: Interpolation von Slices, so dass $dx = dy = dz$
 - ▶ zwei mögliche Sichten auf die Daten
 - ▶ ein Datenwert entspricht einem **Voxel**
 - ▶ 8 Datenwerte an den Ecken einer **Zelle** (quader-/würfelförmig)
- ▶ ... kurvilineare bzw. unstrukturierte Gitter
 - ▶ Daten organisiert als Tetraeder oder Hexaeder
 - ▶ oft: Umwandlung in Tetraeder



3D Volumendaten

Computertomographie-Daten

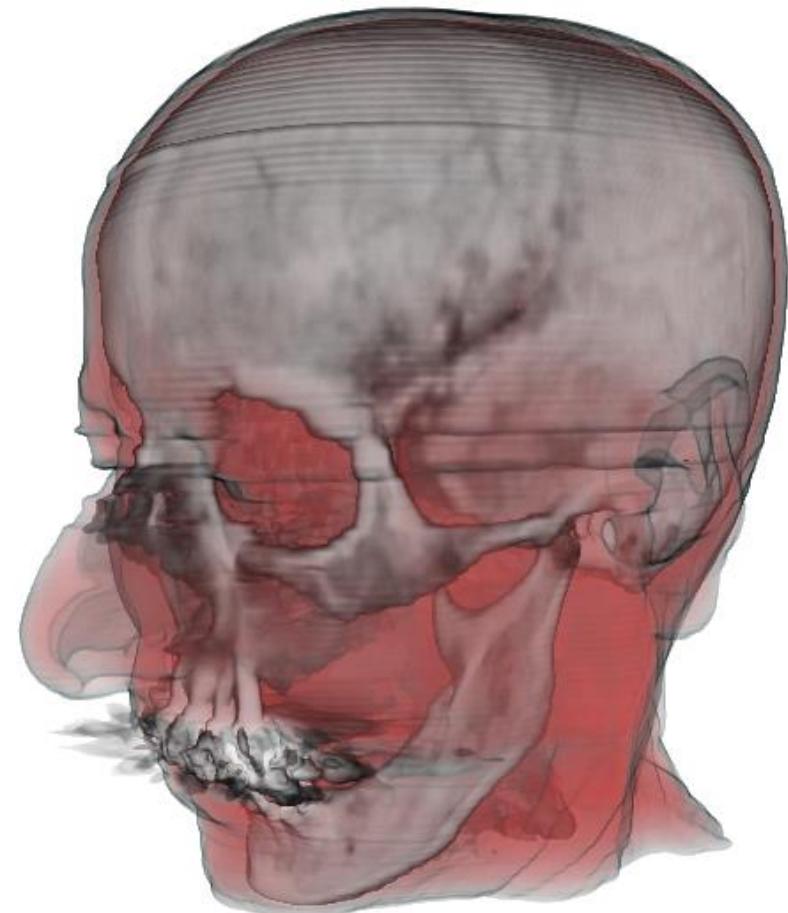
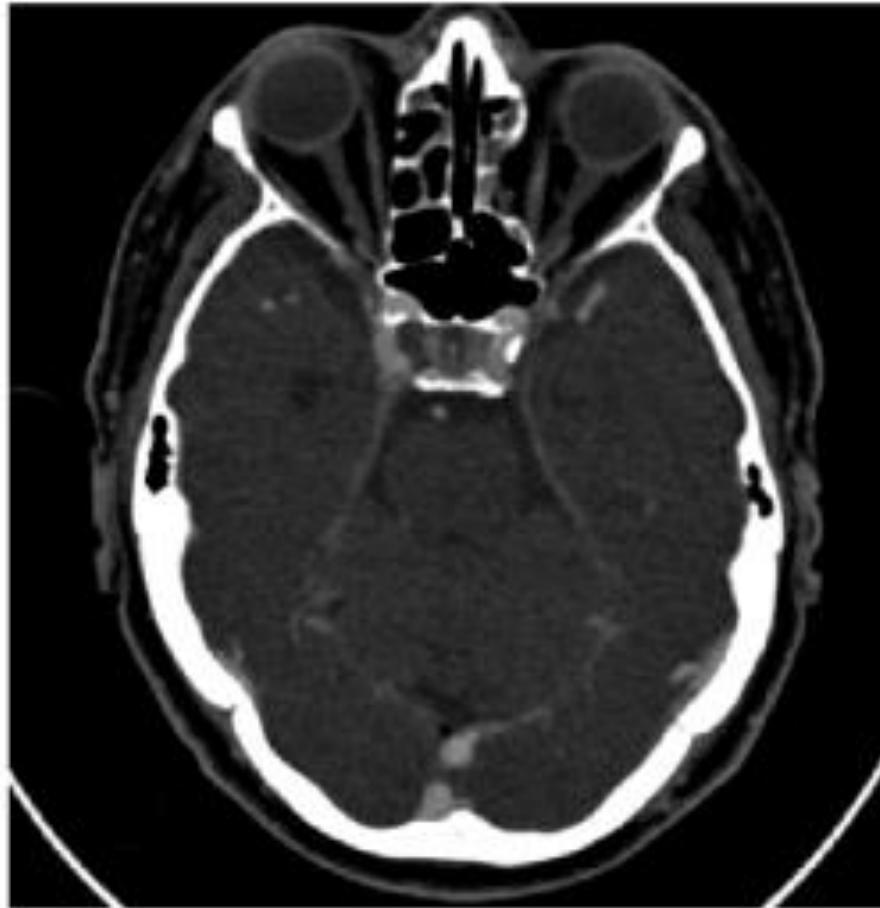
- ▶ Stapel von Schichtbildern, typische Kennzahlen:
 - ▶ Anzahl Schichtbilder: 100 - 2000
 - ▶ Schichtbildgröße: 512^2 bis 1024^2 Pixel
 - ▶ Datenrepräsentation: 12 bis 16 Bits
 - ▶ typische Voxelgröße klinischer Bereich:
 $0.4 \times 0.4 \times 0.4 \text{ mm}^3$, $0.7 \times 0.7 \times 1.2 \text{ mm}^3$
 - ▶ Voxelgröße technischer Bereich:
 abhängig von der Objektgröße, bis zu
 0.005mm (Mikro-CT)
 - ▶ guter Kontrast zwischen Luft,
 Fett- und Muskelgewebe, Knochen
 - ▶ Metallobjekte sind problematisch,
 aber möglich (z.B. Titaniumimplantate)
 - ▶ Zusammenhang zw. Messwerten
 und Gewebeart



3D Volumendaten

Computertomographie-Daten

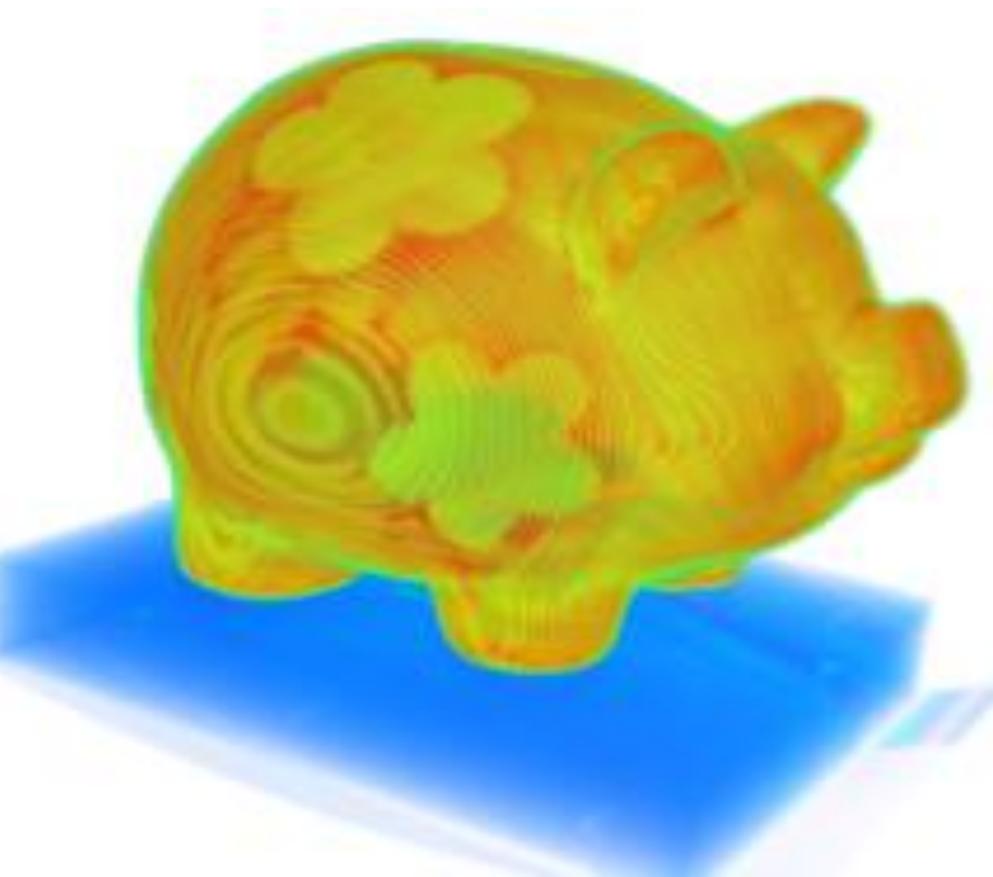
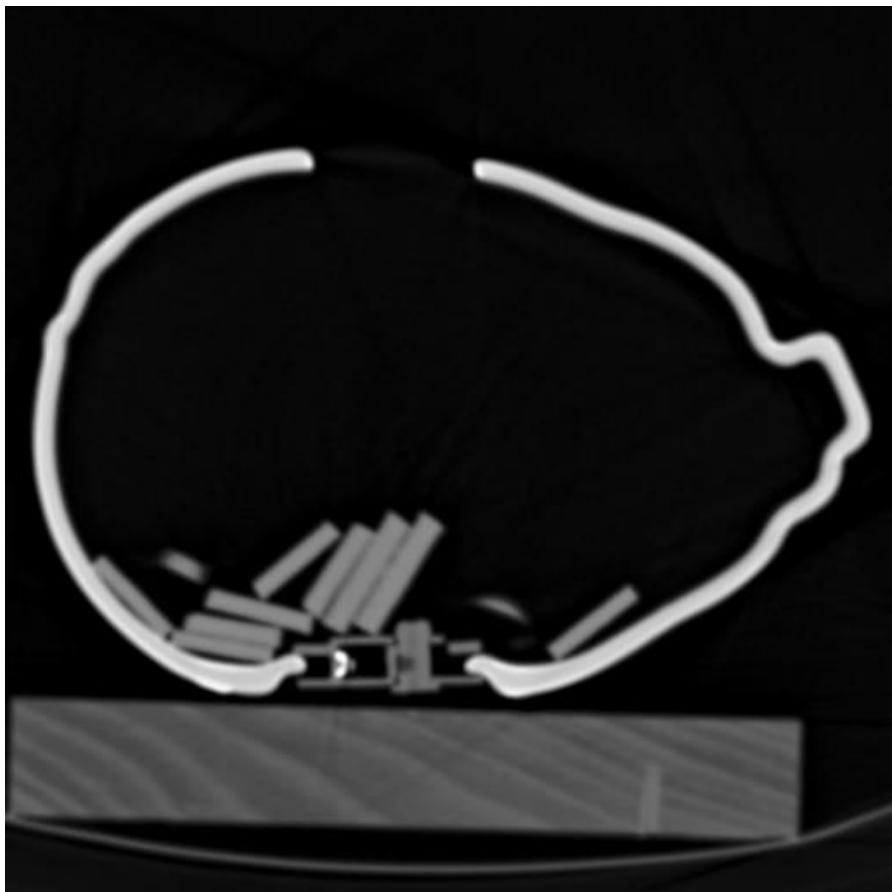
► Beispiele



3D Volumendaten

Computertomographie-Daten

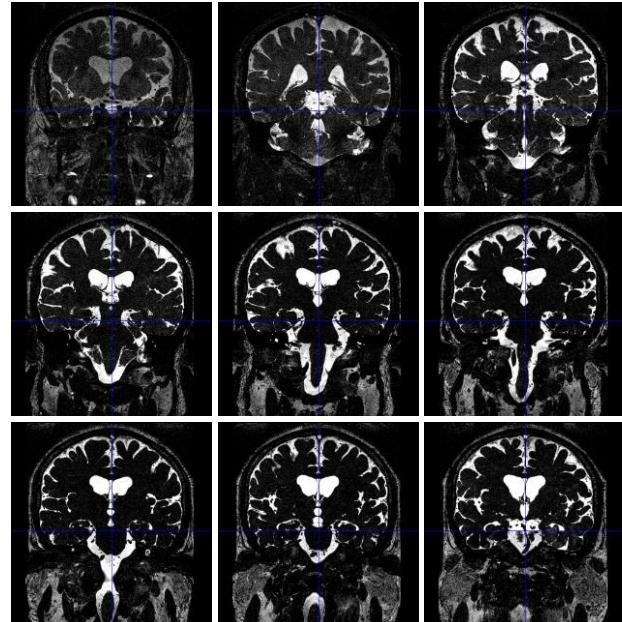
► Beispiele



3D Volumendaten

Magnetresonanztomographie (MRT)

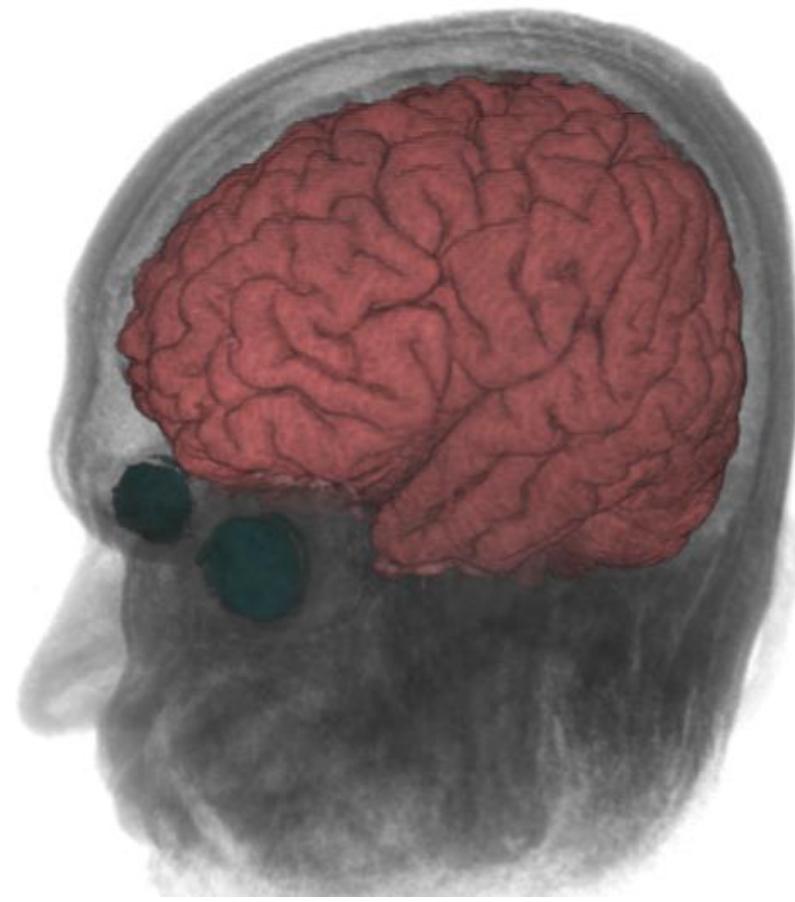
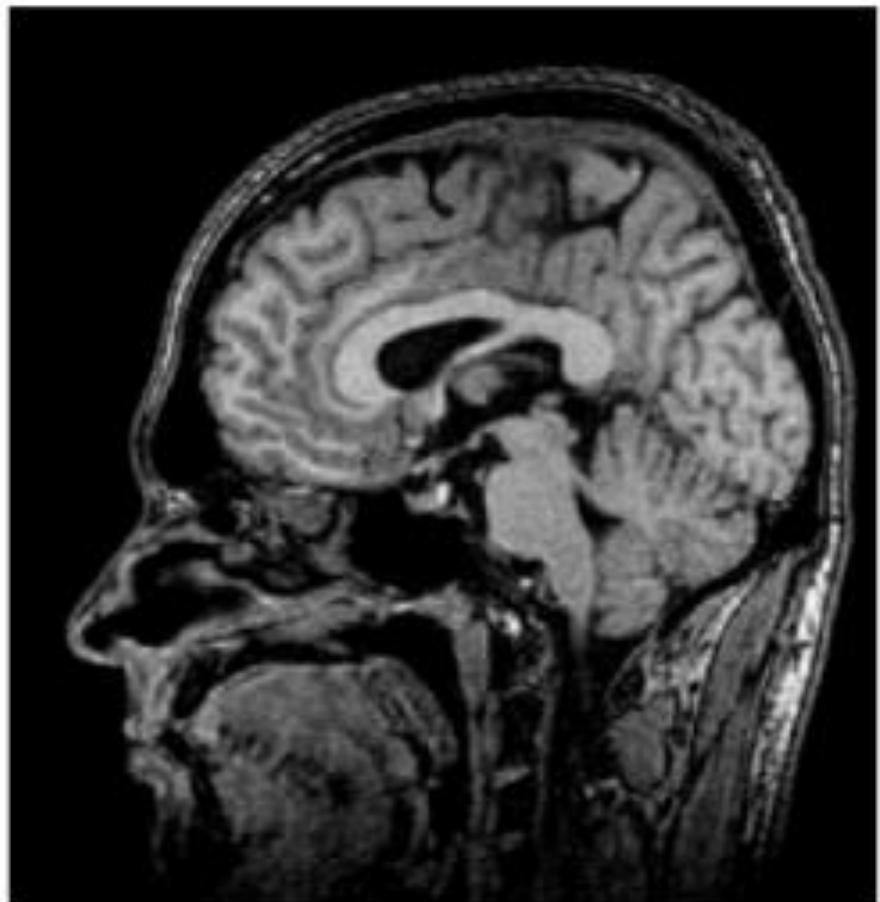
- ▶ Stapel von Schichtbildern, typische Kennzahlen:
 - ▶ Schichtbilder: 40 - 250
 - ▶ Schichtbildgröße: 512^2 bis 1024^2 Pixel
 - ▶ Datenrepräsentation: 12 bis 16 Bits
 - ▶ Voxelgröße (klinischer Bereich):
 $0.7 \times 0.7 \times 0.7 \text{ mm}^3$, $1.0 \times 1.0 \times 1.7 \text{ mm}^3$
 - ▶ Forschungsbereich: bis 0.02mm
 - ▶ keine ionisierende Strahlung:
 - ▶ guter Kontrast bei weichem Gewebe, Knochen erscheinen dunkel
 - ▶ Metallobjekte können nicht gescannt werden
 - ▶ kein direkter Zusammenhang zw. Messwert und Gewebetyp



3D Volumendaten

Magnetresonanztomographie (MRT)

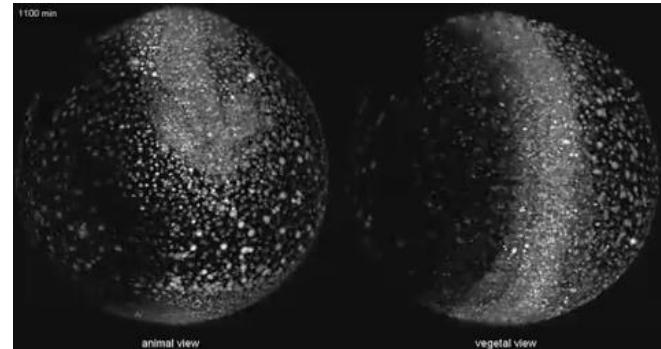
► Beispiele



Volumenvisualisierung

Herausforderungen

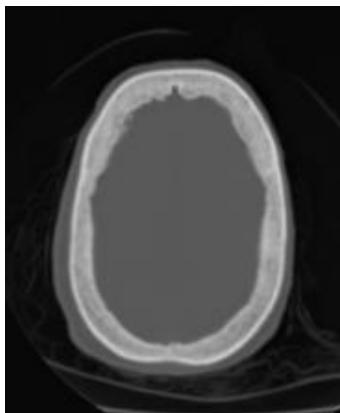
- ▶ große Datenmengen
 - ▶ $1024^2 \times 512$ Voxel à 16 Bit = 1 Gigabyte
 - ▶ Zeitabhängigkeit
 - ▶ Multimodalität: Druck, Temperatur, Dichte, Phase, Absorption, ...
 - ▶ im Gegensatz zu CT/MRT-Messungen sind Simulationsdaten heute „beliebig groß“
- ▶ Darstellung durch Projektion auf ein 2D Bild
 - ▶ viel Information, wenige Pixel: wie stelle ich nur das Wichtigste dar?
 - ▶ z.B. 2^{29} Voxel auf 1920×1080 Pixel (259:1)
- ▶ Geschwindigkeit
 - ▶ Interaktion ist sehr wichtig, daher mindestens 10 Bilder/Sek.



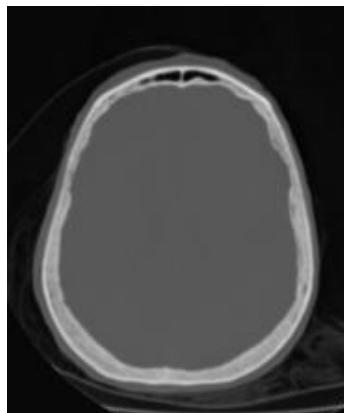
Slicing – 2D Ansatz

Orthogonales Slicing (... eine Form der Selektion, also Filtering-Stufe)

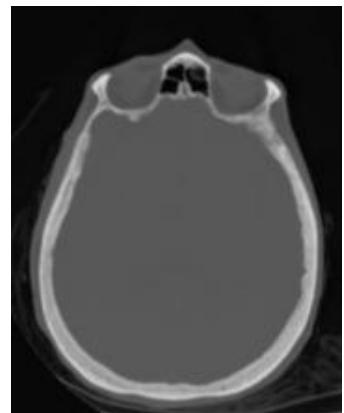
- ▶ stelle einzelne Schichtbilder senkrecht zur x -, y -, z -Achse aus den Volumendaten nebeneinander dar (keine technische Herausforderung)
- ▶ Abbildung der Skalarwerte auf gut unterscheidbare Farben (gleich)
- ▶ Mediziner: „die Wahrheit liegt im Schichtbild“



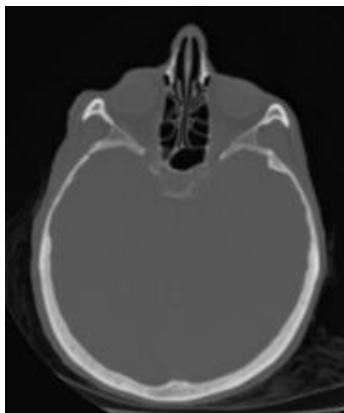
Slice 20



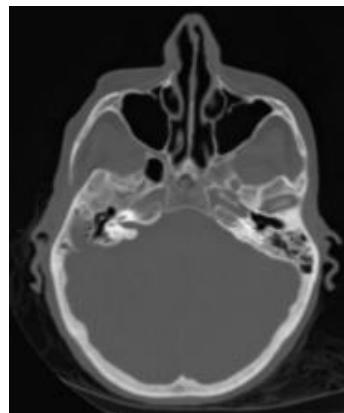
30



40



50



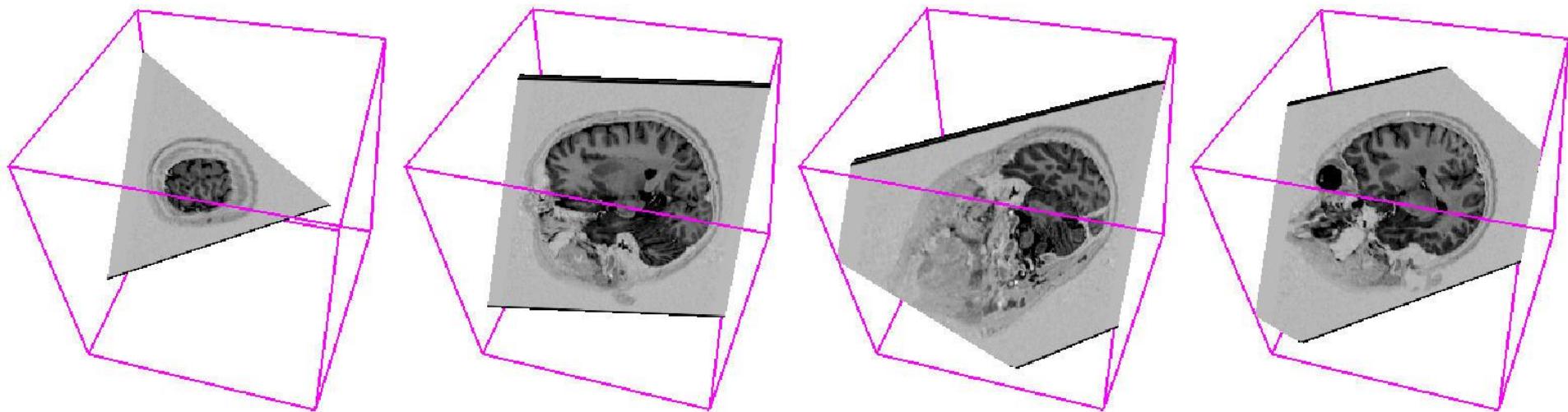
60

CT Datensatz

Slicing – 2D Ansatz

Oblique Slicing (... eine Form der Selektion, also Filtering-Stufe)

- ▶ Abtasten der Daten auf einer beliebig orientierten Ebene
- ▶ Darstellung auf der GPU (auch keine technische Herausforderung):
 - ▶ speichere Volumendaten in einer 3D Textur
 - ▶ berechne Schnittpolygon der Ebene und Bounding Box des Volumens
 - ▶ Darstellung: texturiertes Polygon mit trilinearer Interpolation



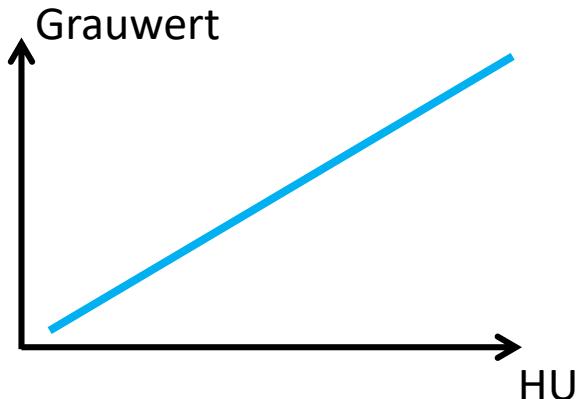
Klassifikation

- ▶ Ziele der Klassifikation
 - ▶ gestattet dem Benutzer Strukturen sichtbar zu machen
 - ▶ extrahiert wichtige Features aus dem Datensatz
- ▶ Klassifikation mittels **Transferfunktionen**
 - ▶ die nächsten Beispiele zeigen die Anwendung für Schichtbilder, später für volumetrische Darstellung
 - ▶ Farbtabelle: **Abbildung der Skalarwerte auf Grauwert oder Farbe**
- ▶ Klassifikation ist keine triviale Aufgabe
 - ▶ Histogramme geben wertvolle Hinweise
 - ▶ oft werden Transferfunktionen interaktiv verändert
- ▶ unabdingbar für Volumenvisualisierung und Schichtbilder

Klassifikation: Transferfunktionen für Schichtbilder

Beispiel

- ▶ Rampenfunktion: $f_1: [-1000; 3000] \rightarrow [0,255]$
- ▶ alle Datenwerte abgebildet, aber keine Details sichtbar



Transferfunktionen

Hounsfield-Skala

- ... beschreibt die Abschwächung von Röntgenstrahlung, basierend auf dem Schwächungskoeffizient μ (physikalische Größe): $I = I_0 \cdot e^{-\mu d}$

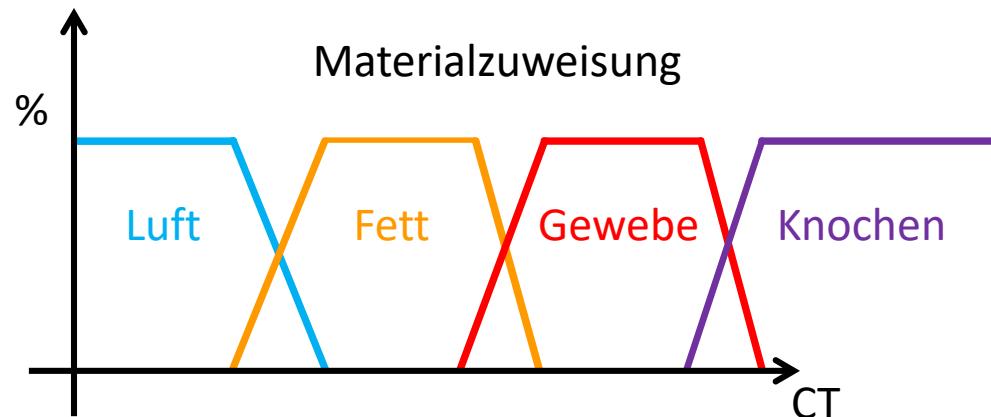
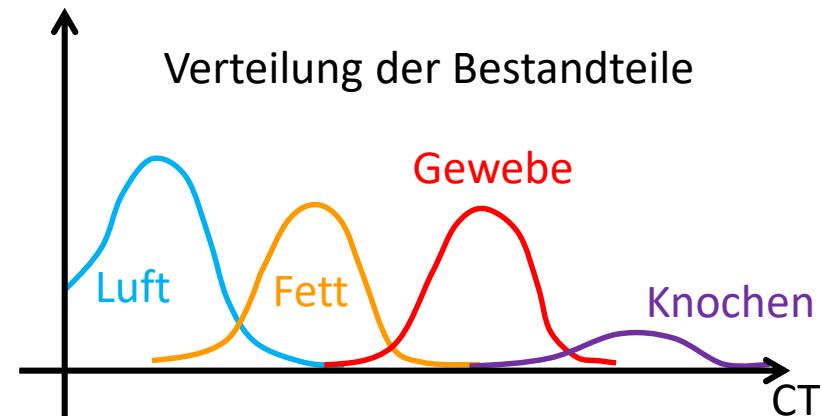
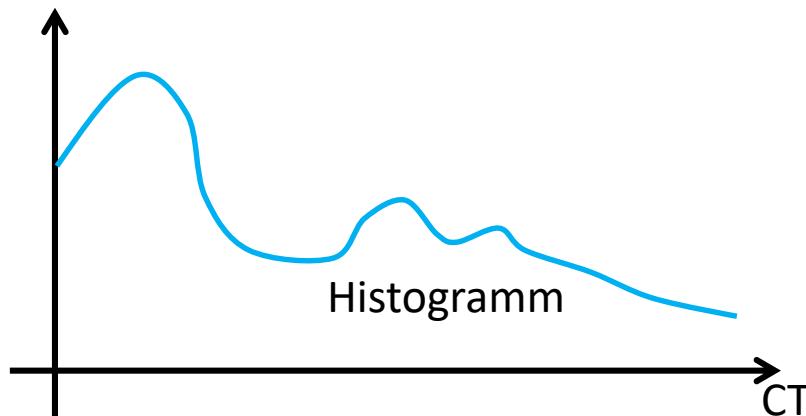
CT-Zahl von Gewebe: $\frac{\mu_{Gewebe} - \mu_{Wasser}}{\mu_{Wasser}} \cdot 1000 \text{ HU}$ (Hounsfield-Units)

- meist verwendet man 12-Bit CT-Messungen, die den Bereich von -1024 bis $+3071 \text{ HU}$ abdecken
 - Luft: -1024 HU
 - Fett -100 HU bis -20 HU
 - Wasser 0 HU
 - weiches Gewebe (z.B. Muskeln) $+20 \text{ HU}$ bis $+80 \text{ HU}$
 - Knochen: $> +500 \text{ HU}$
 - Metalle können sehr viel größere Werte aufweisen!
- für Visualisierung werden die 12-Bit Daten oft auf 8-Bit reduziert
 - i.d.R. durch Windowing, d.h. auswählen eines Teilintervalls und dementsprechend verbunden mit Verlust von Dynamikumfang

Transferfunktionen

Wie gewinnt man Transferfunktionen? (hier für CT)

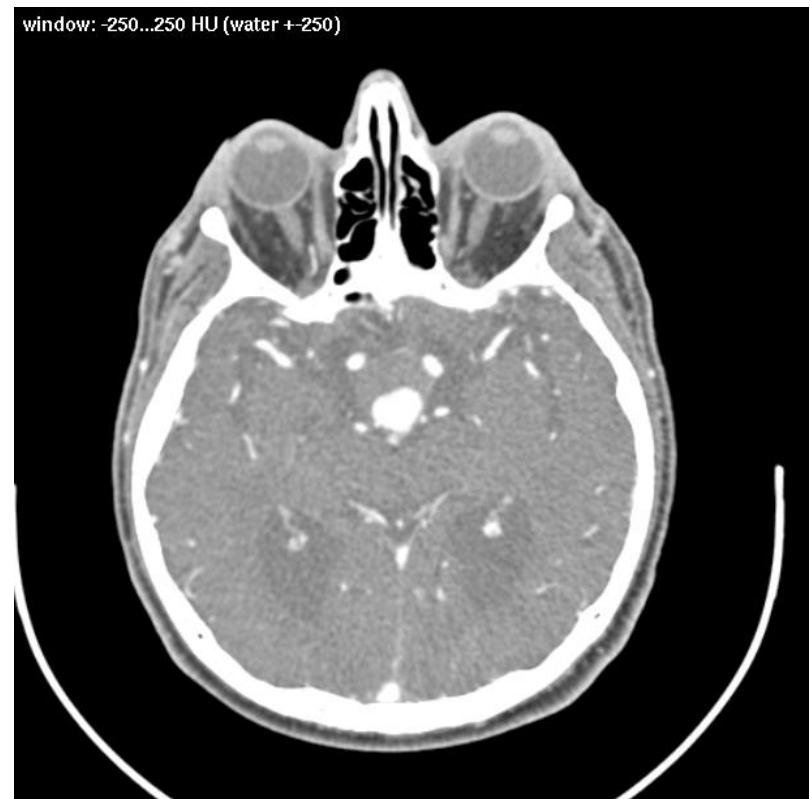
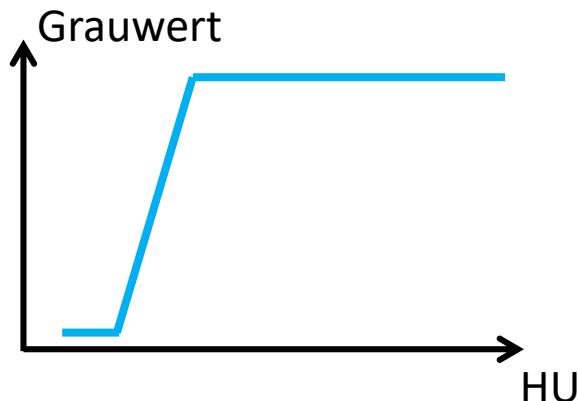
- ▶ im Prinzip ein heuristischer Ansatz
- ▶ basierend auf Messungen und vielen Datensätzen



Klassifikation: Transferfunktionen für Schichtbilder

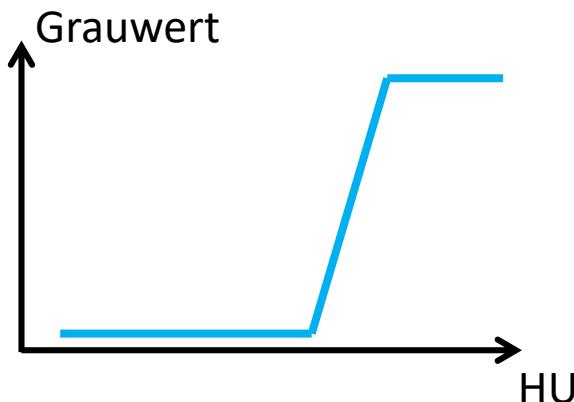
Beispiel

- ▶ Window-Funktion: $f_1: [-250; 250] \rightarrow [0,255]$
- ▶ mittlerer Wert: 0 (Wasser), Breite der Rampe: 500
- ▶ Blutgefäße werden hervorgehoben



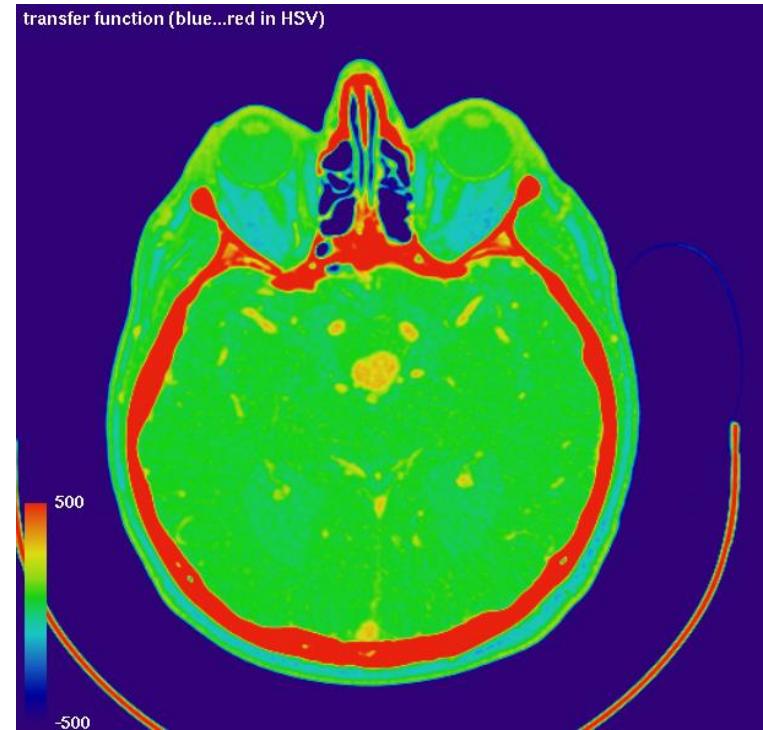
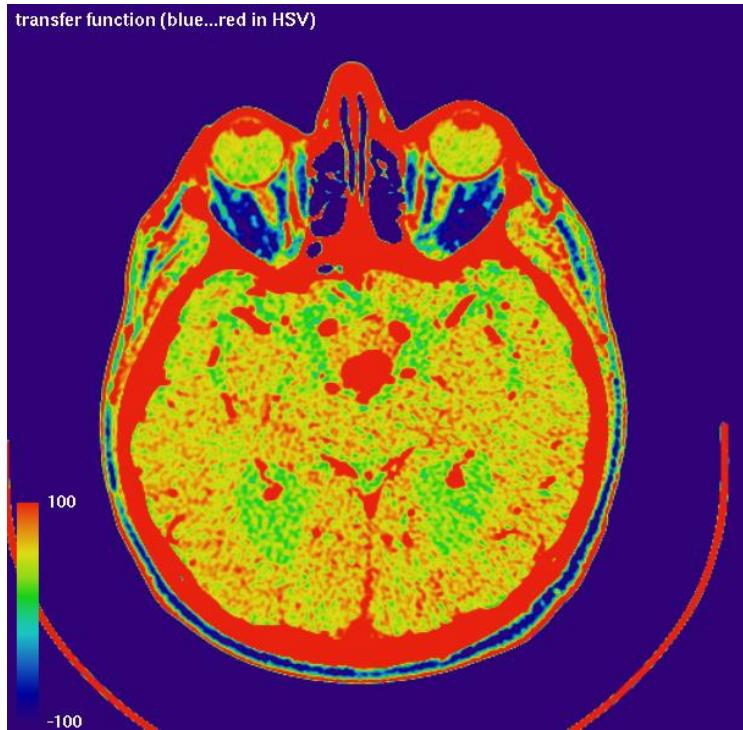
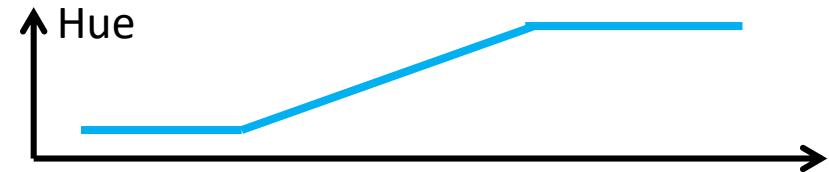
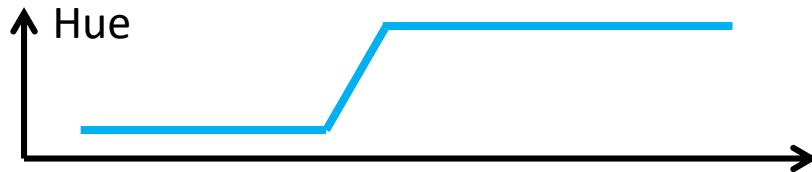
Beispiel

- ▶ Window-Funktion: $f_3: [750; 1250] \rightarrow [0,255]$
- ▶ mittlerer Wert: 1000 (Knochen), Breite der Rampe: 500
- ▶ Knochenstruktur wird dargestellt



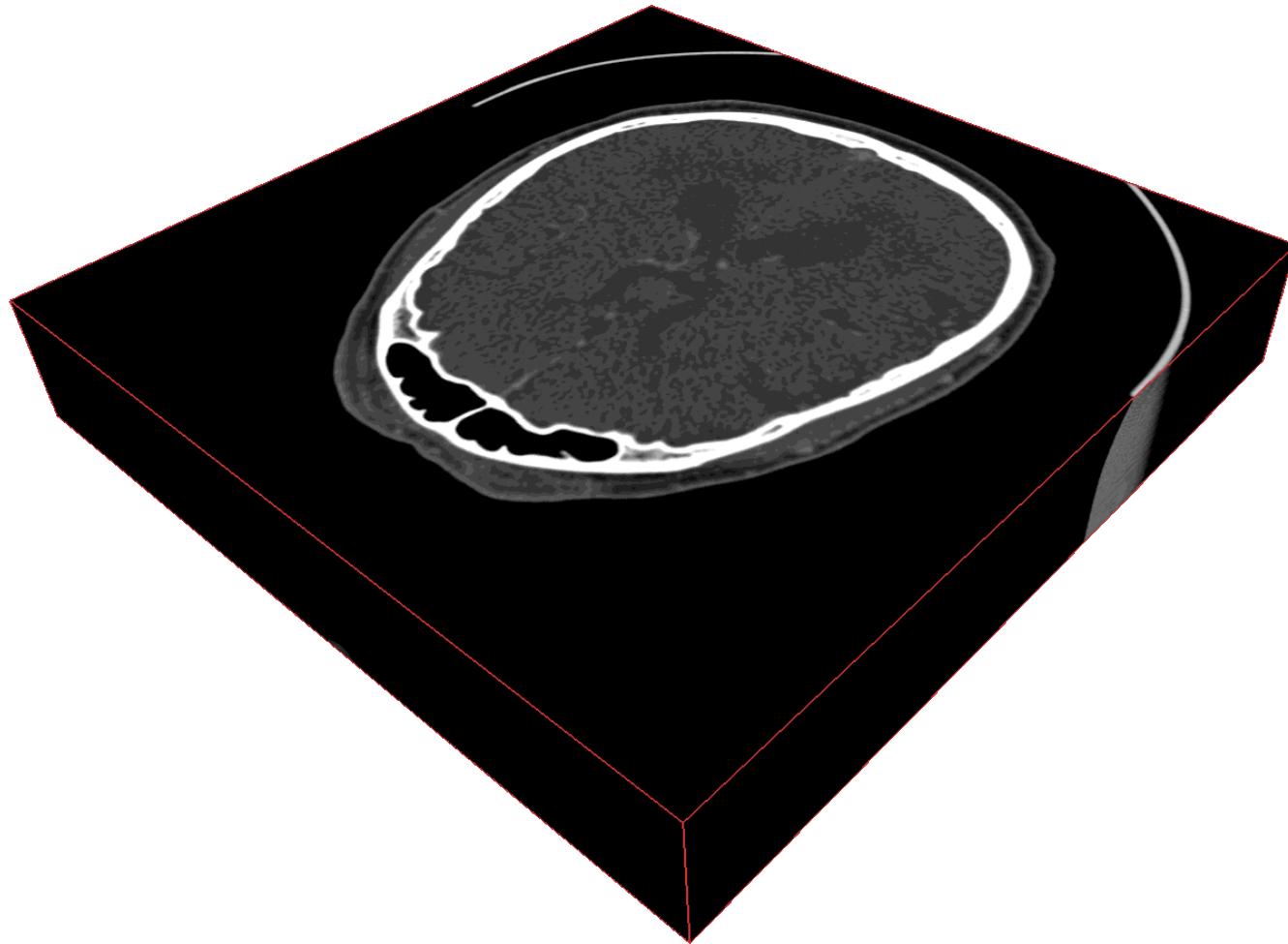
Klassifikation: Transferfunktionen für Schichtbilder

Beispiel: Abbildung auf Hue-Wert im HSV Farbmodell



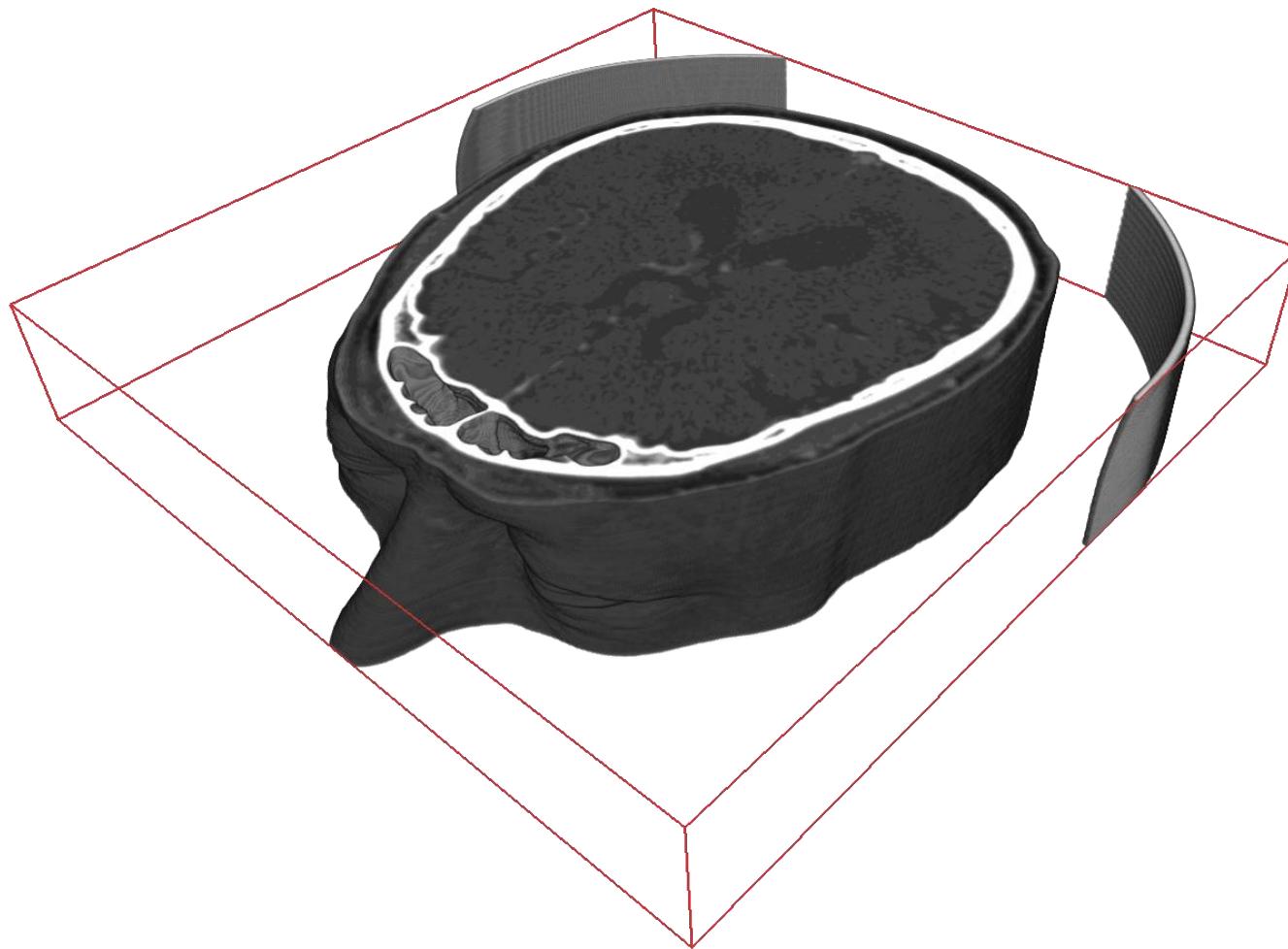
Transferfunktionen

Beispiel: Klassifikation für volumetrische Darstellung



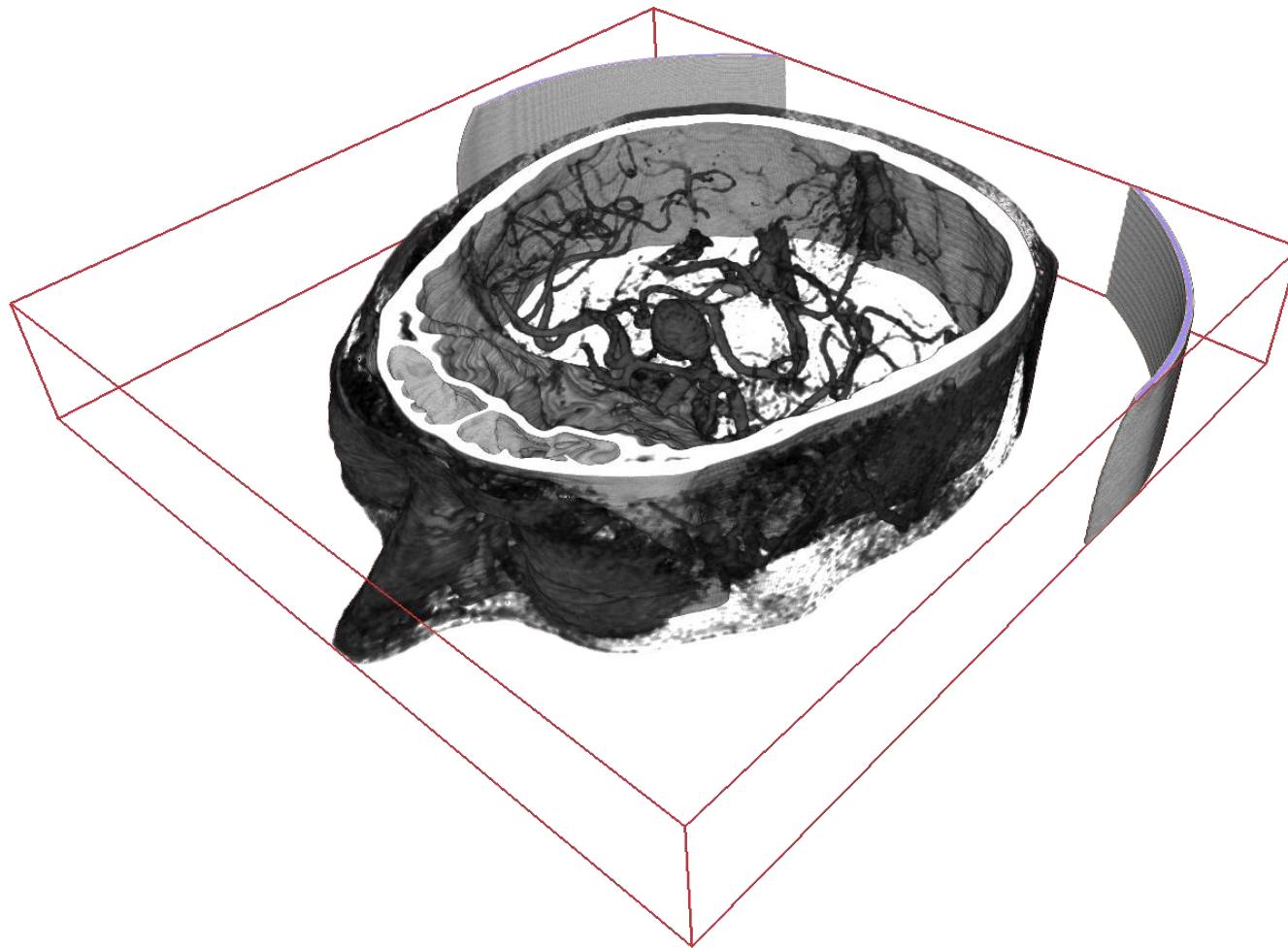
Transferfunktionen

Beispiel: Klassifikation für volumetrische Darstellung



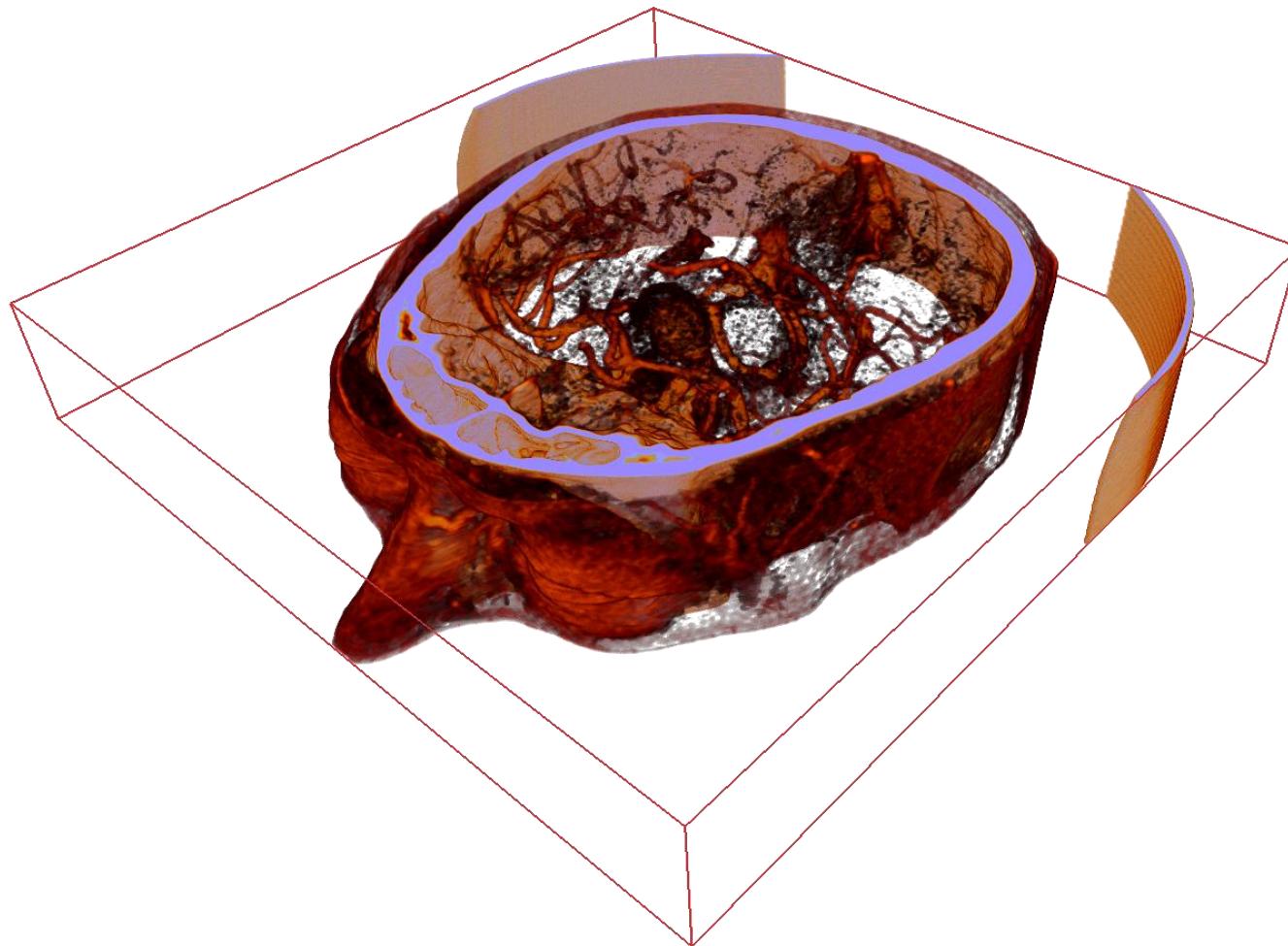
Transferfunktionen

Beispiel: Klassifikation für volumetrische Darstellung



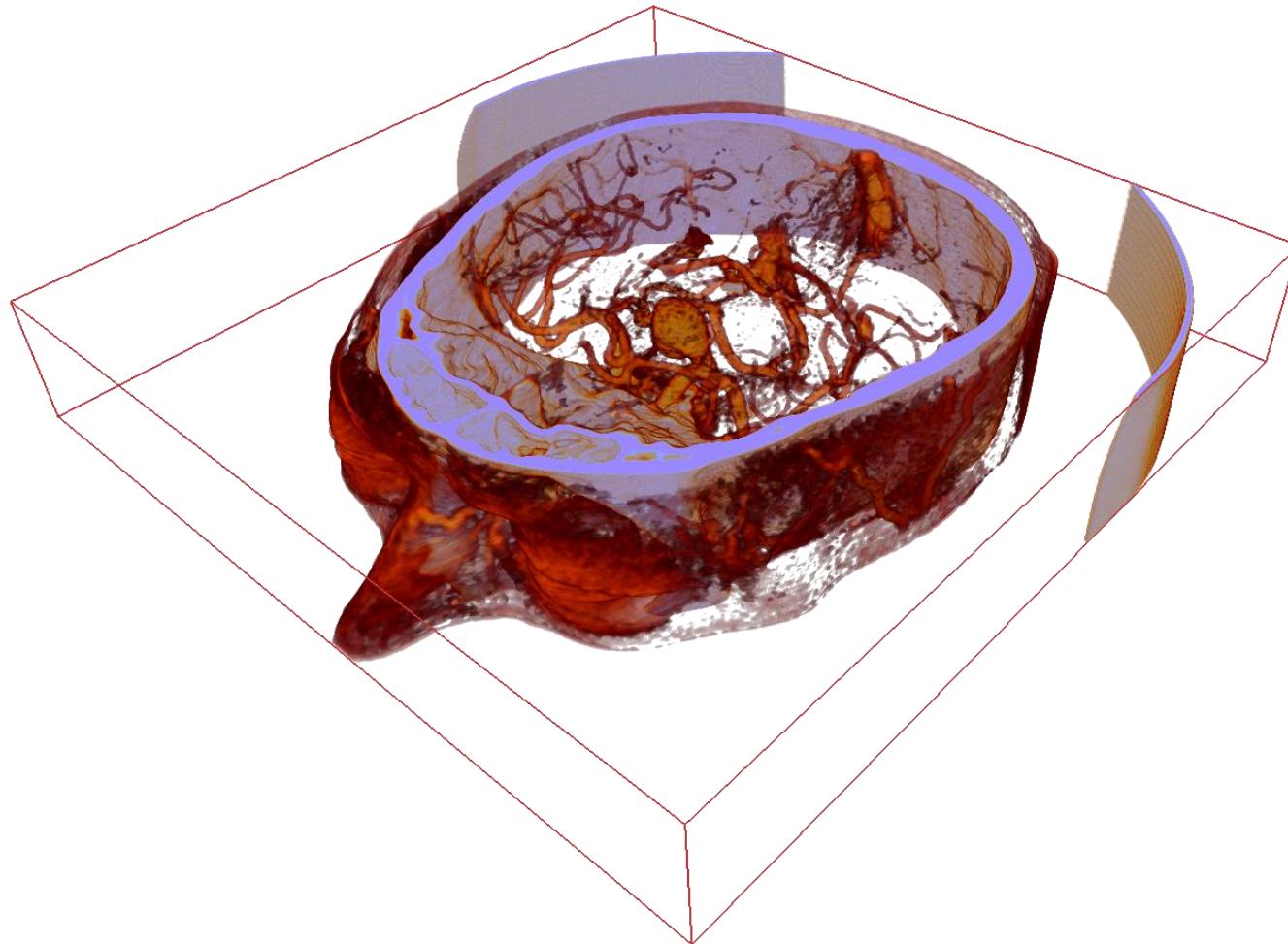
Transferfunktionen

Beispiel: Klassifikation für volumetrische Darstellung



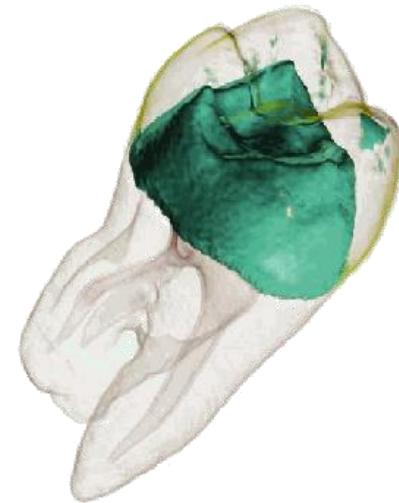
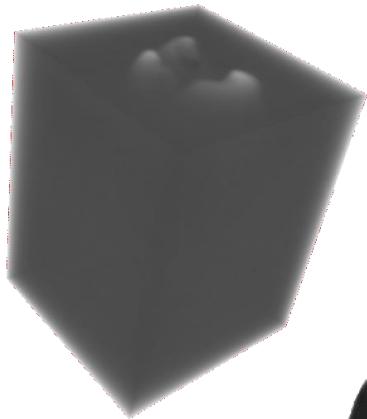
Transferfunktionen

Beispiel: Klassifikation für volumetrische Darstellung



Direkte Volumenvisualisierung

Beispiele: unterschiedliche Transferfunktionen



Volumenvisualisierung

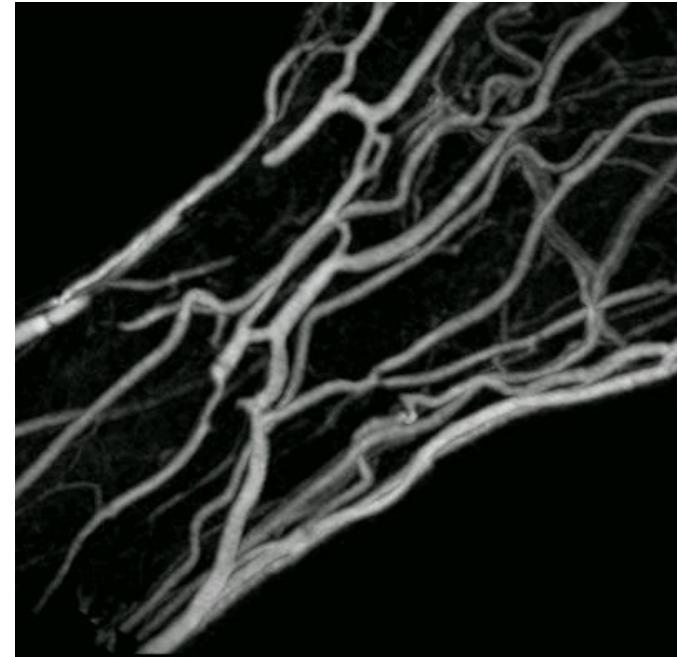
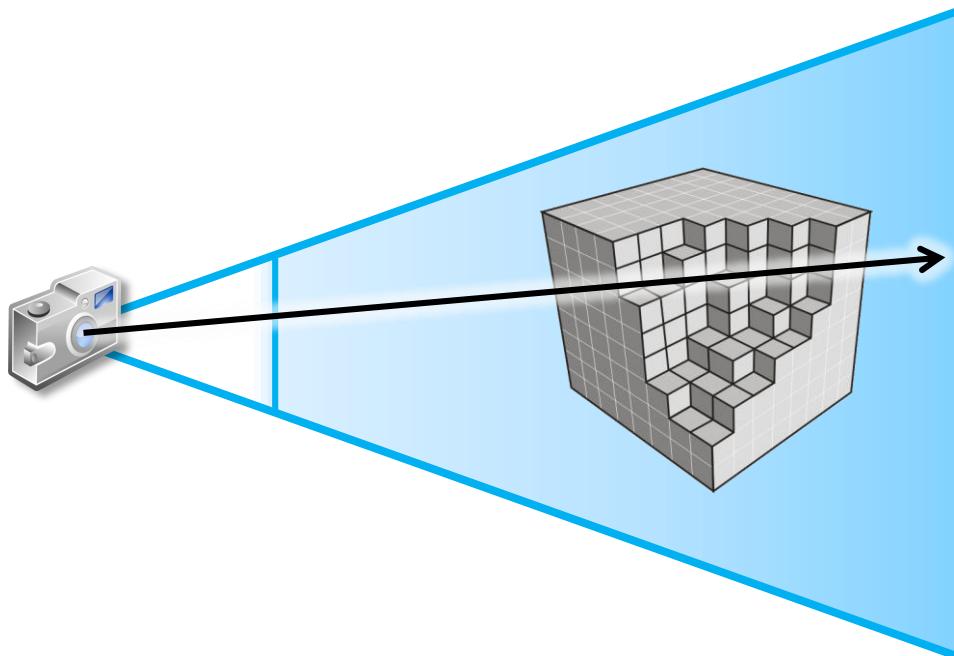
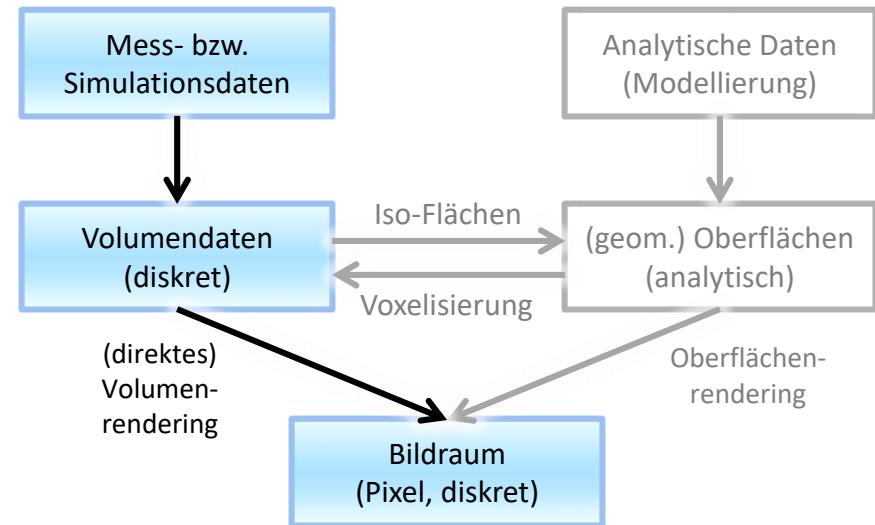
- ▶ Einführung zu Volumenvisualisierung
 - ▶ Volumendaten
 - ▶ einfache Methoden (Slicing, ...)
 - ▶ Oberflächen- vs. Volumenrendering
 - ▶ Überblick über die Techniken
- ▶ Direkte Volumenvisualisierung
 - ▶ Transferfunktionen
 - ▶ Einführung, Kombinationsmöglichkeiten
 - ▶ ...
- ▶ Indirekte Volumen Visualisierung

Oberflächen- vs. Volumenrendering

Zusammenhänge und Begriffe

Beispiel

- ▶ MR-Messung
- ▶ Maximum Intensity Projection:
Bild zeigt Blutgefäße in einer
Hand (mit Kontrastmittel)

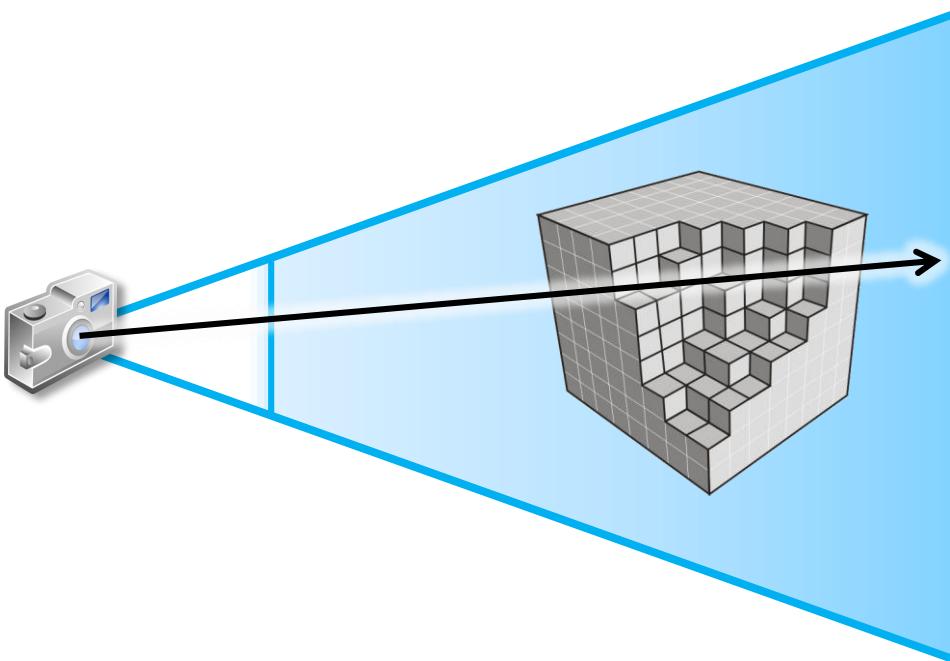
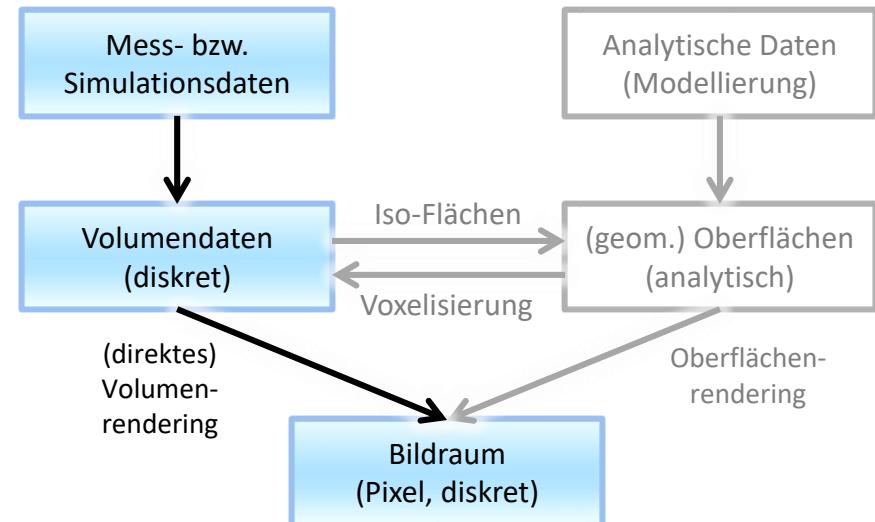


Oberflächen- vs. Volumenrendering

Zusammenhänge und Begriffe

Beispiel

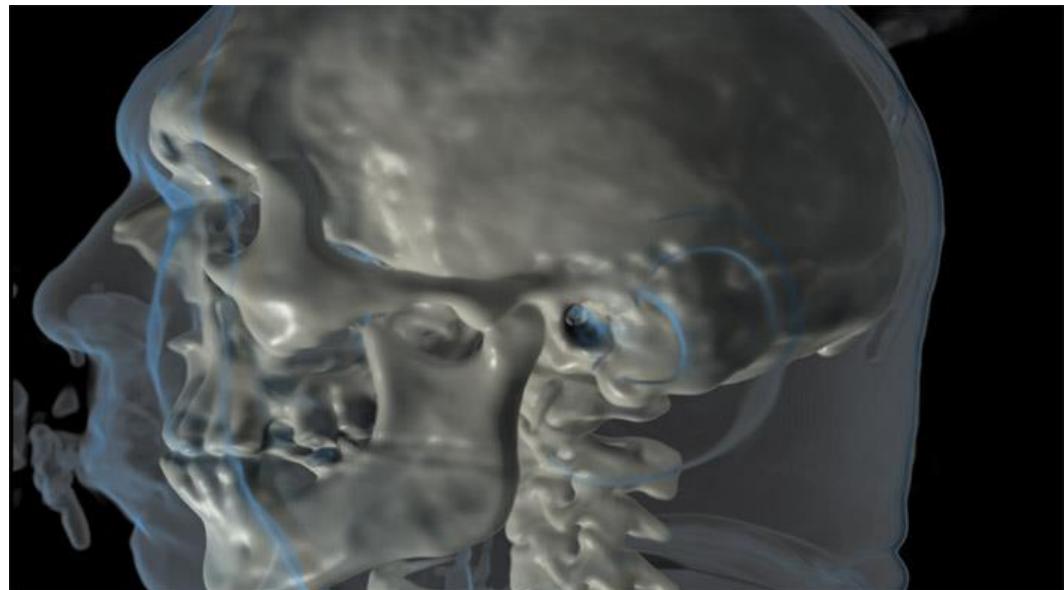
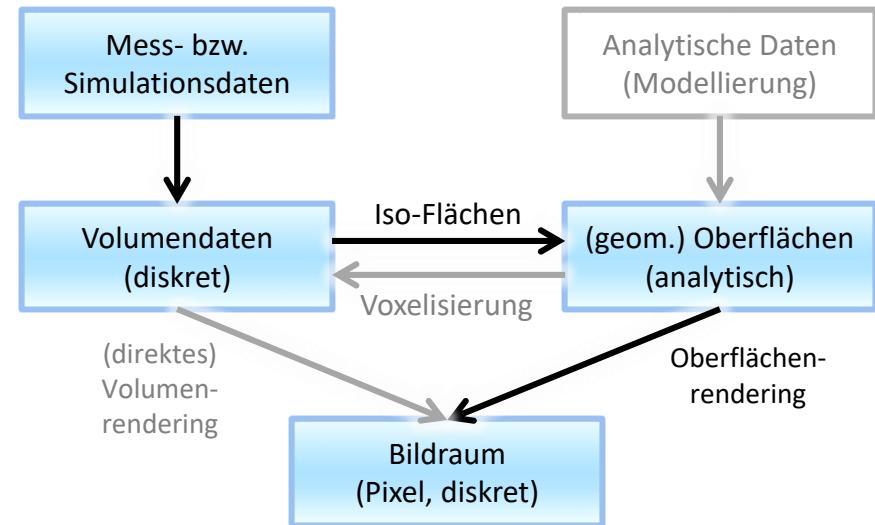
- ▶ CT-Messung
- ▶ Maximum Intensity Projection:
Animation zeigt Knochenstruktur
einer Maus



Oberflächen- vs. Volumenrendering

Zusammenhänge und Begriffe

- ▶ Beispiel
 - ▶ CT-Messung
 - ▶ Berechnung von Isoflächen (Marching Cubes)
 - ▶ Oberflächenrendering (OpenGL)

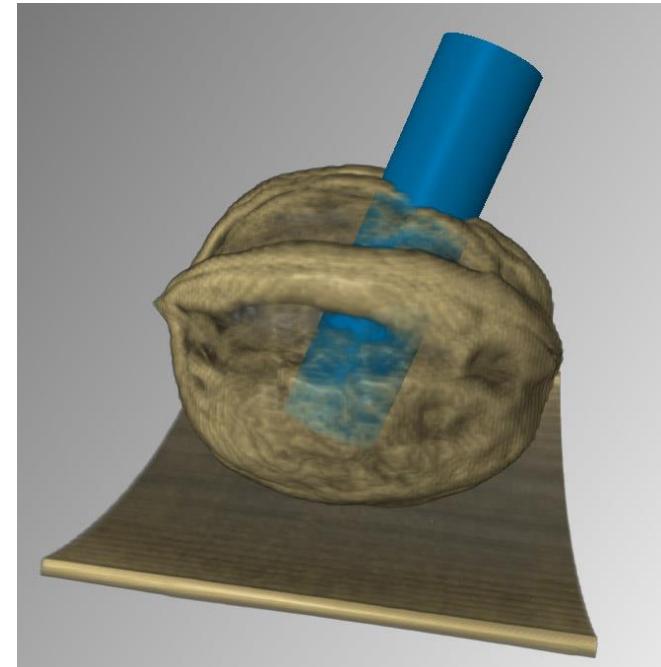
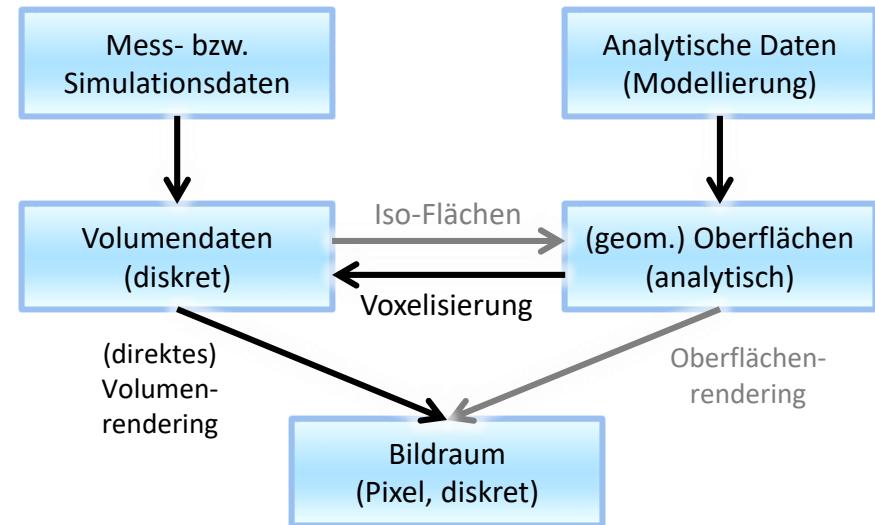


Oberflächen- vs. Volumenrendering

Zusammenhänge und Begriffe

Beispiel

- ▶ CT-Daten
- ▶ Definition einer Oberfläche
- ▶ Abtastung (Voxelisieren) und Kombination der Daten
- ▶ direktes Volumenrendering (semitransparente Bereiche)

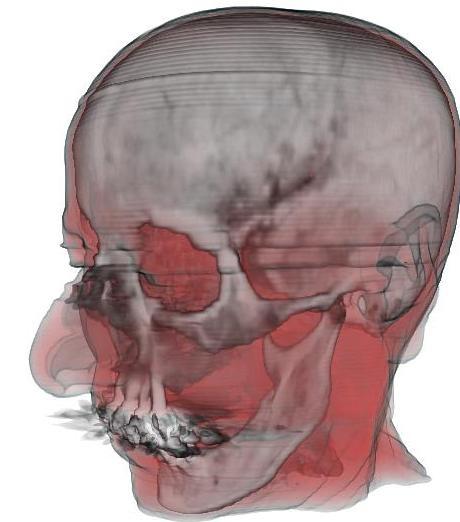


Oberflächen- vs. Volumenrendering

Zusammenhänge und Begriffe

- ▶ Oberflächenrendering
 - ▶ **indirekte** Volumenvisualisierung
 - ▶ Zwischenrepräsentation: Iso-Fläche
(oder andere 3D Fläche)
 - ▶ Vorteil: einfaches und schnelles
Rendering mit Grafikhardware

- ▶ Volumenrendering
 - ▶ **direkte** Volumenvisualisierung
 - ▶ Klassifikation mittels Transferfunktionen
 - ▶ „Farbtabelle“ für Volumenvisualisierung
 - ▶ Abbildung der Skalarwerte und
Gradienten auf Farbe, Opazität, Emission, ...
 - ▶ Vorteile: stellt innere Strukturen durch (Semi-)Transparenz dar,
keine Zwischenrepräsentation, „echte“ 3D Strukturen



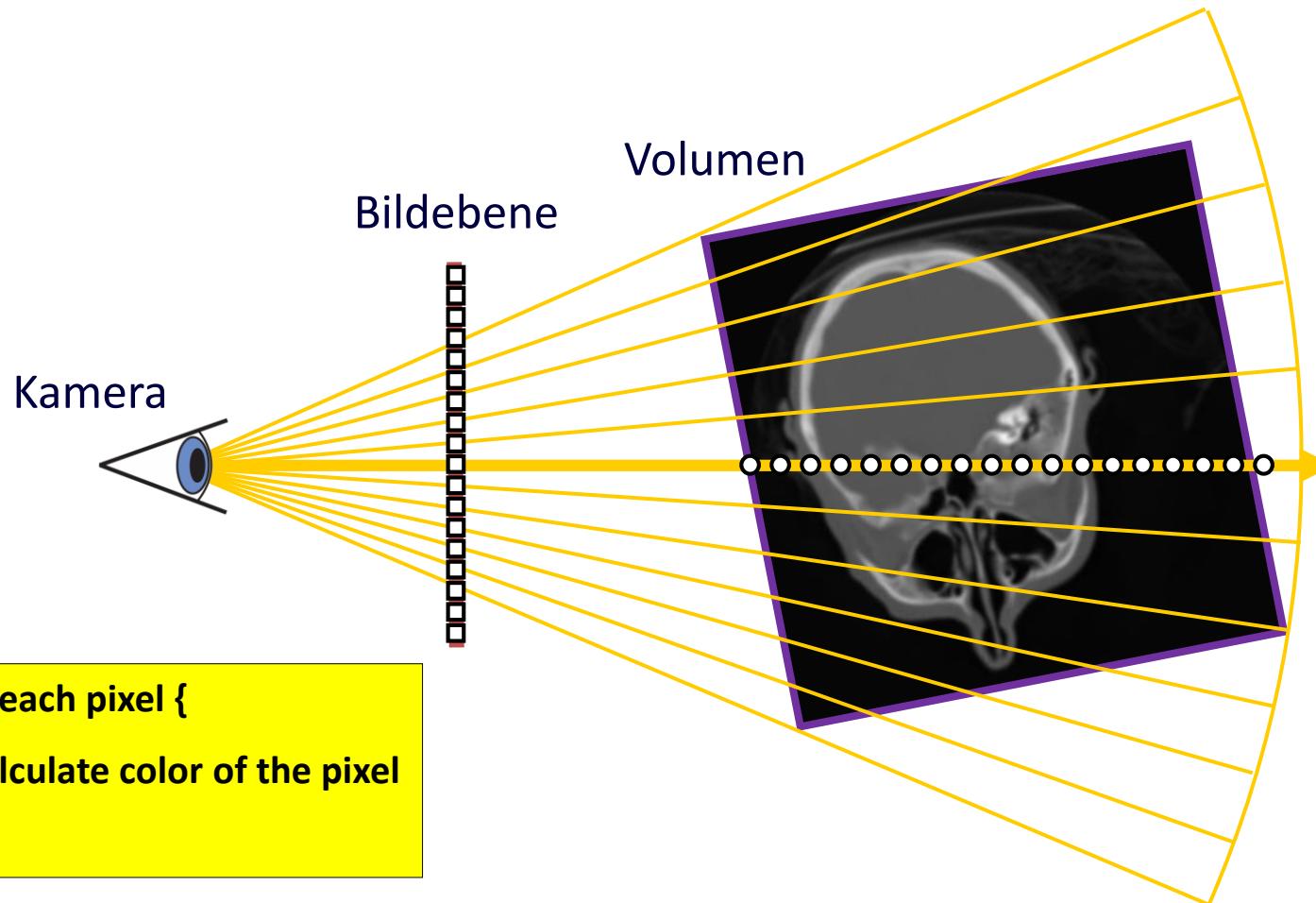
Volumenvisualisierung

- ▶ Einführung zu Volumenvisualisierung
 - ▶ Volumendaten
 - ▶ einfache Methoden (Slicing, ...)
 - ▶ Oberflächen- vs. Volumenrendering
 - ▶ Überblick über die Techniken
- ▶ **Direkte Volumenvisualisierung**
 - ▶ Darstellungstechniken
 - ▶ Transferfunktionen
 - ▶ ...
- ▶ **Indirekte Volumen Visualisierung**
 - ▶ ...

Direkte Volumenvisualisierung

Image-Order Ansatz: Raycasting bzw. Raymarching

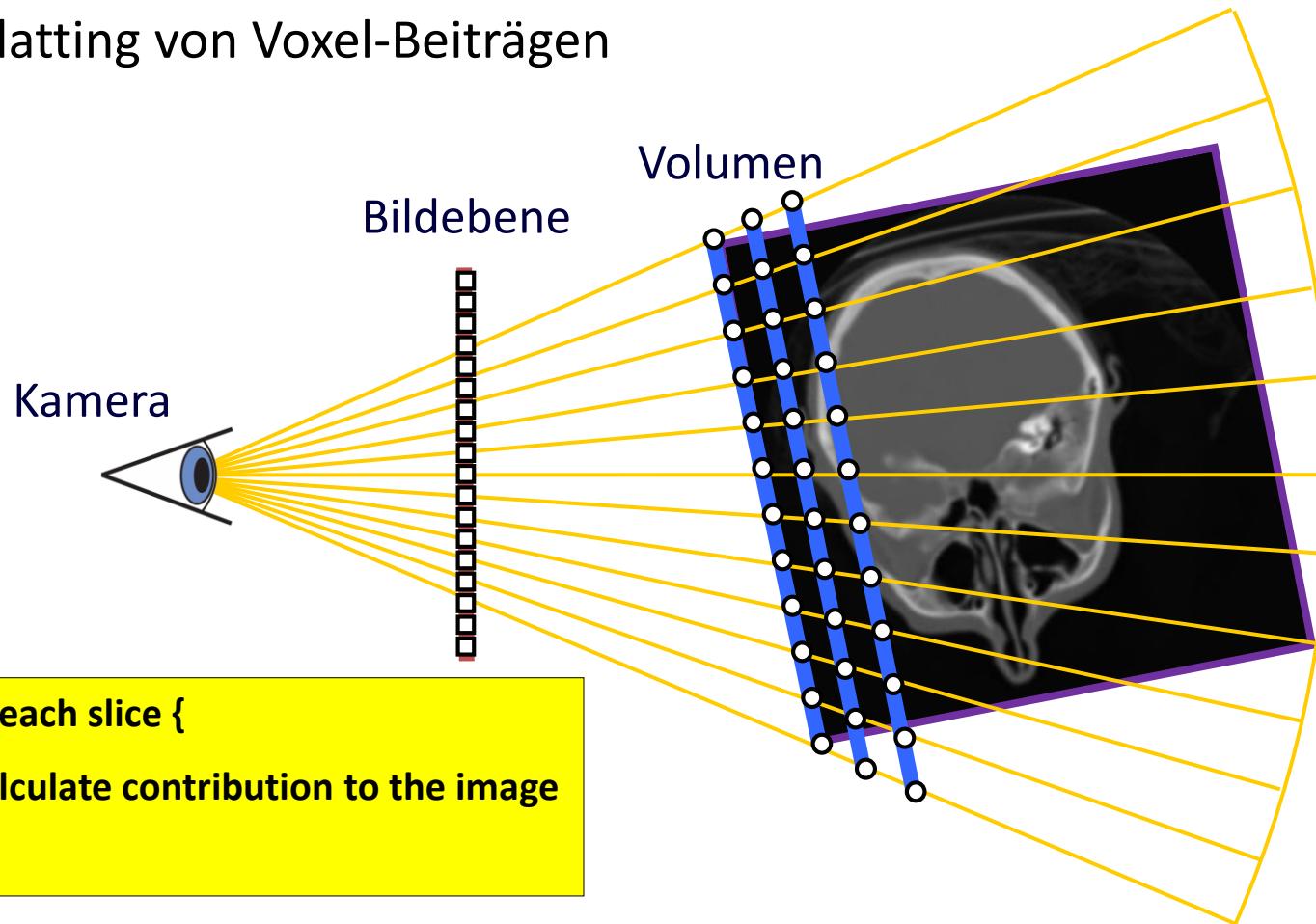
- mit aktueller Grafikhardware die wichtigste Technik



Direkte Volumenvisualisierung

Object-Order Ansatz: Slicing (texturbasiert)

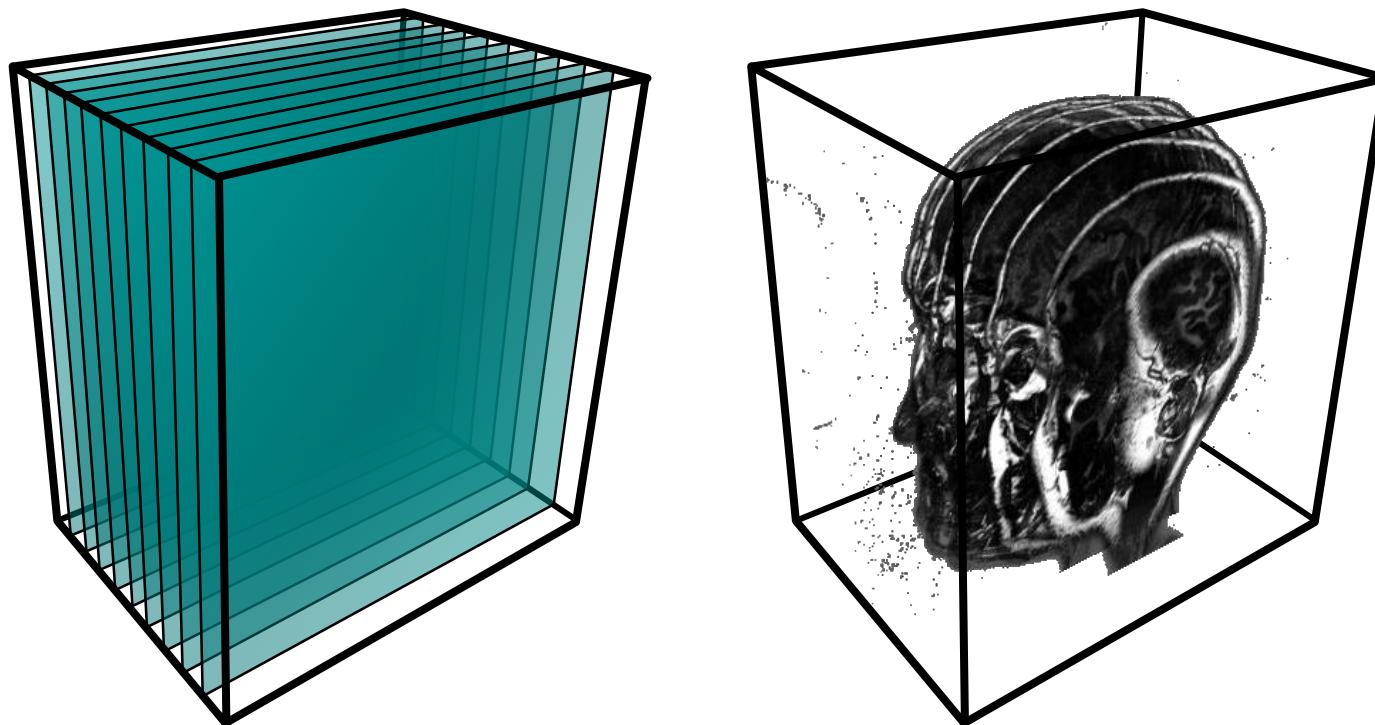
- meist Rendering von Slices, manchmal Splatting von Voxel-Beiträgen



Direkte Volumenvisualisierung

Object-Order Ansatz: Slicing (texturbasiert)

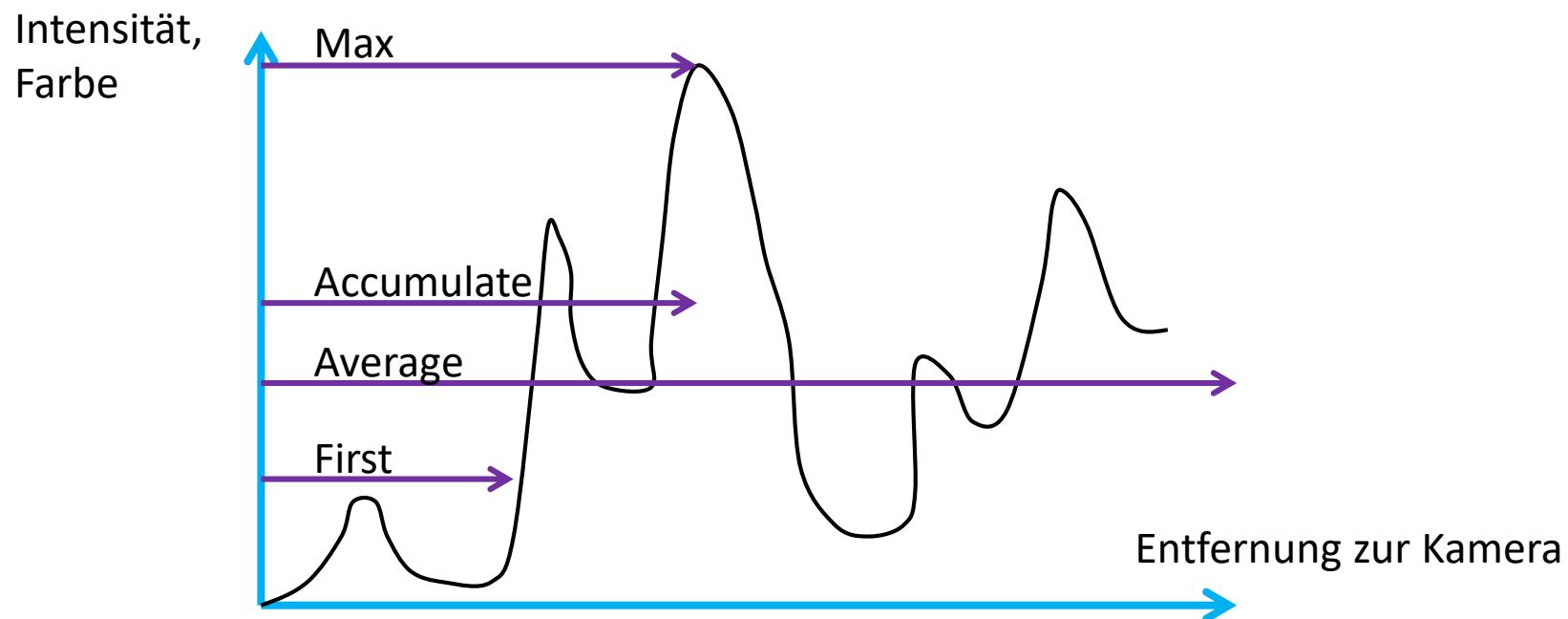
- ▶ als Proxygeometrie werden polygonale Slices gezeichnet



Direkte Volumenvisualisierung

Überblick

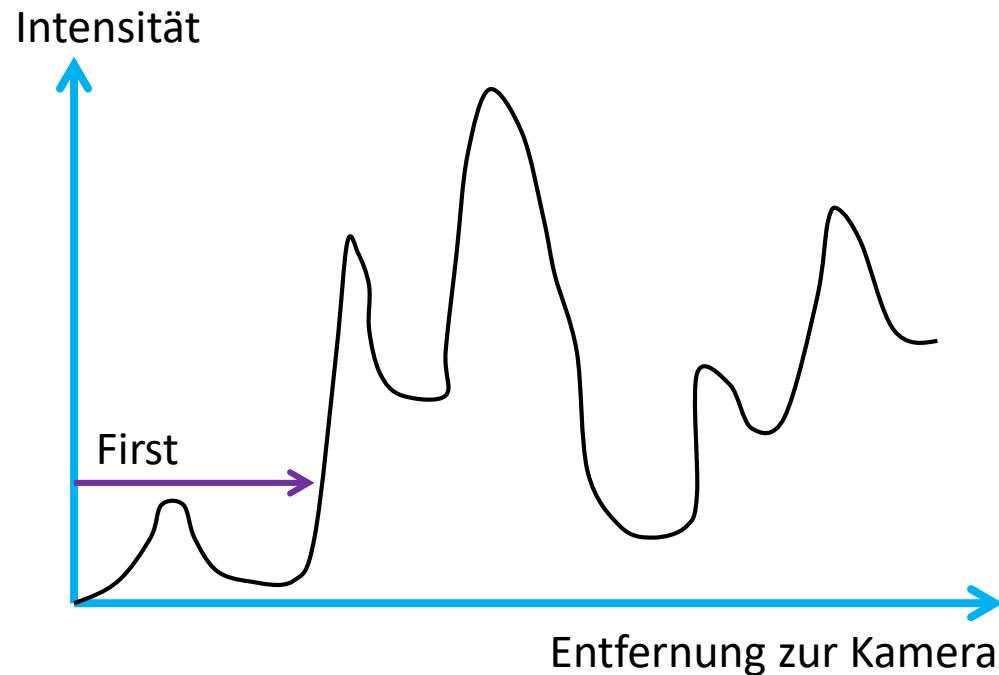
- wie wird die Pixel-Farbe aus den Skalarwerten bestimmt?
 - ▶ **Transferfunktion:** bildet Skalarwerte auf Helligkeit/Farbe/Opazität ab
 - ▶ **Schattierung:** optional
 - ▶ **Compositing:** setzt die Beiträge entlang eines Sichtstrahls zu einer finalen Pixel-Farbe zusammen (z.B. mittels Maximum Intensity Projection, Mittelung, Akkumulation, First Hit, etc.)



Compositing

Compositing: First

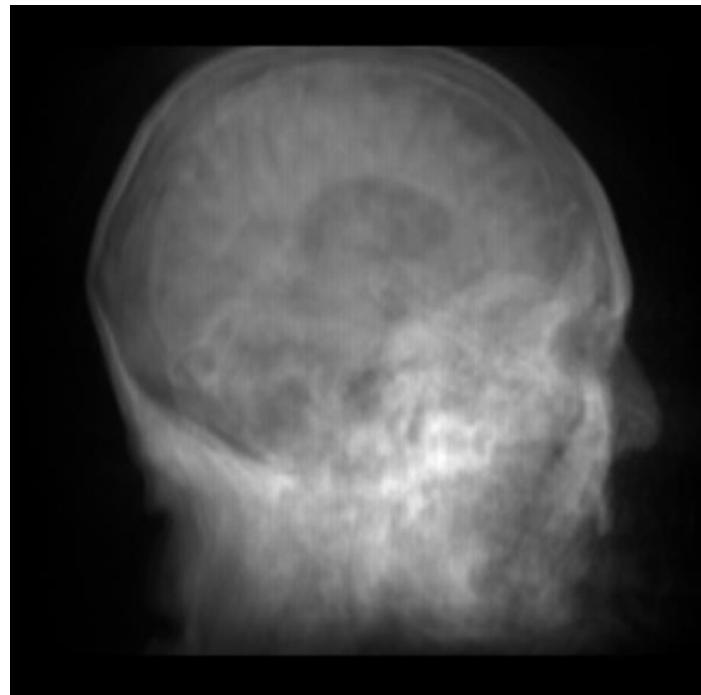
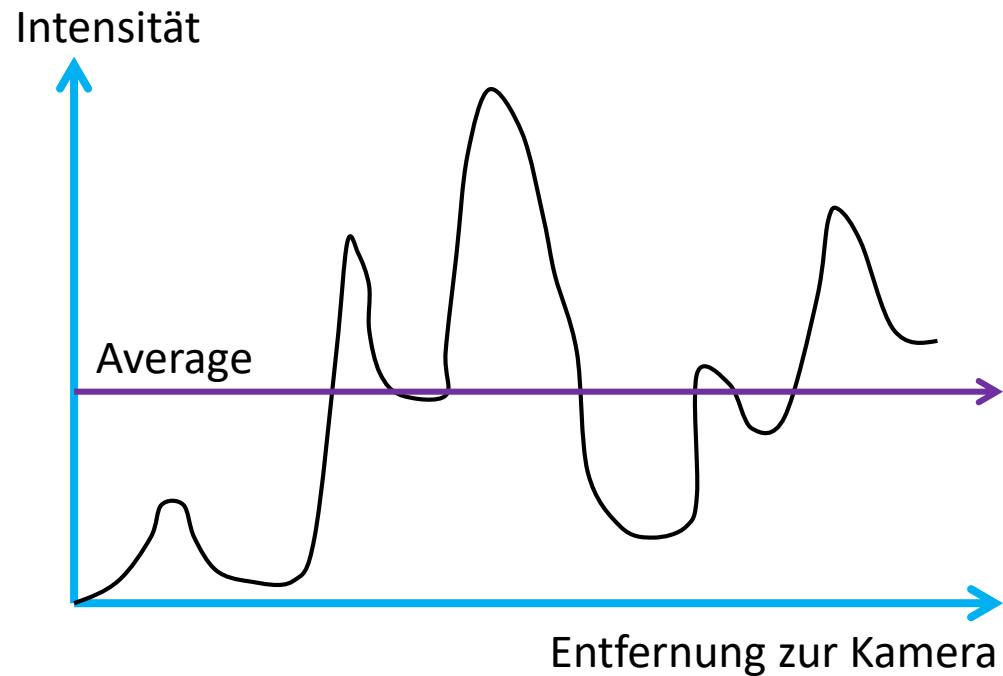
- ▶ stelle nur die Voxel dar, die entlang eines Sichtstrahls als erstes einen festgelegten Schwellwert überschreiben (alle anderen transparent)
- ▶ stellt praktisch Isoflächen dar (die Beleuchtung muss, wie später gezeigt, noch berechnet werden)



Compositing

Compositing: Average

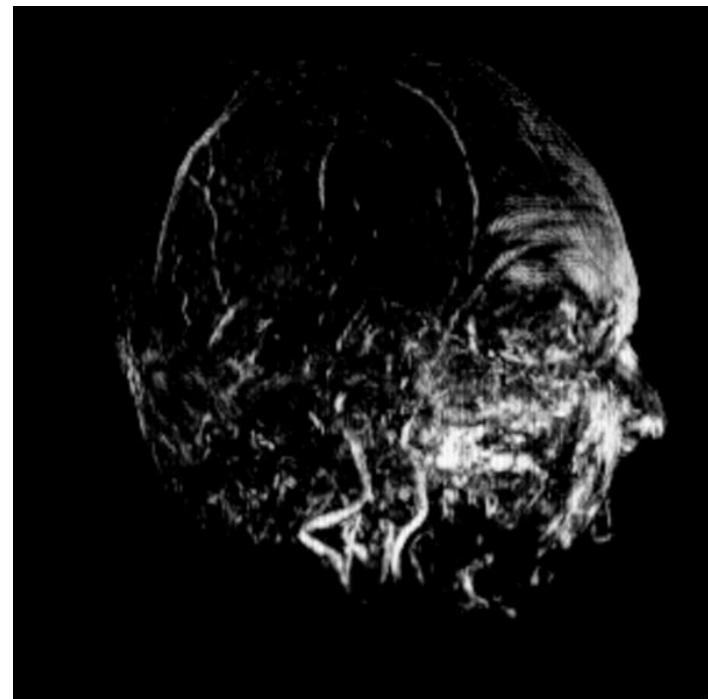
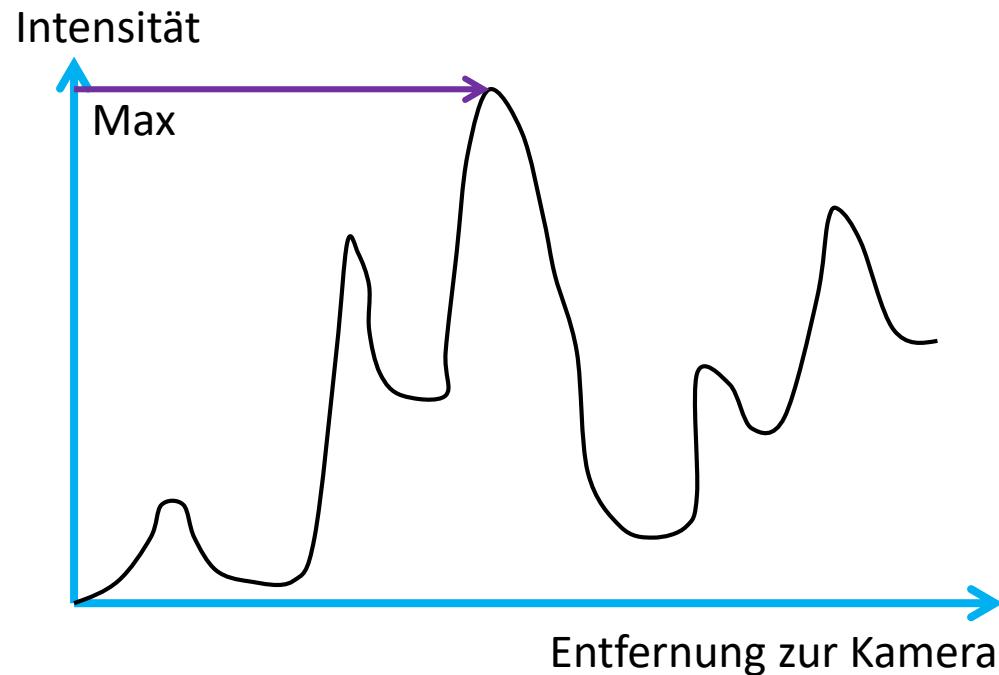
- ▶ mittelt die Skalarwerte entlang eines Sichtstrahls durch das Volumen
- ▶ erzeugt im Prinzip ein Röntgenbild (bei CT-Daten natürlich)



Compositing

Compositing: Maximum Intensity Projection (MIP)

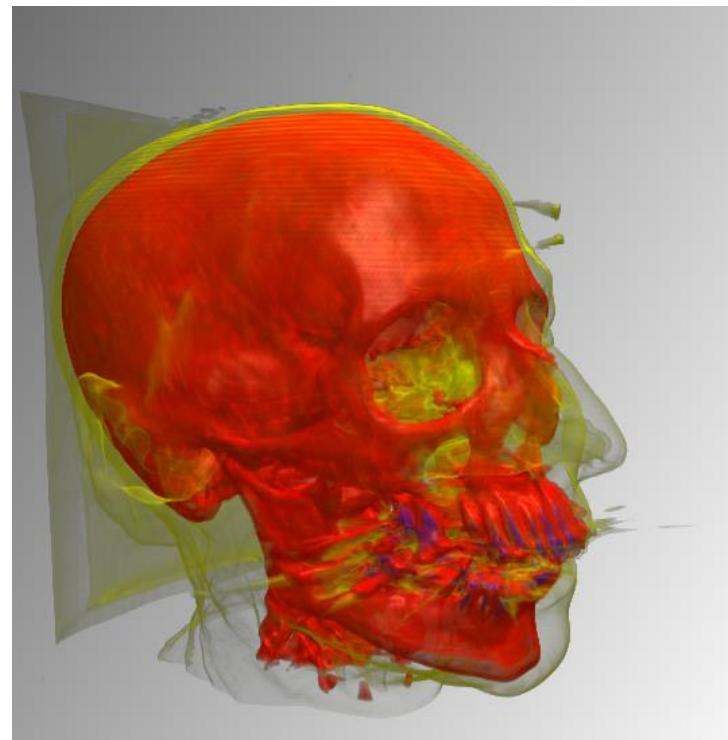
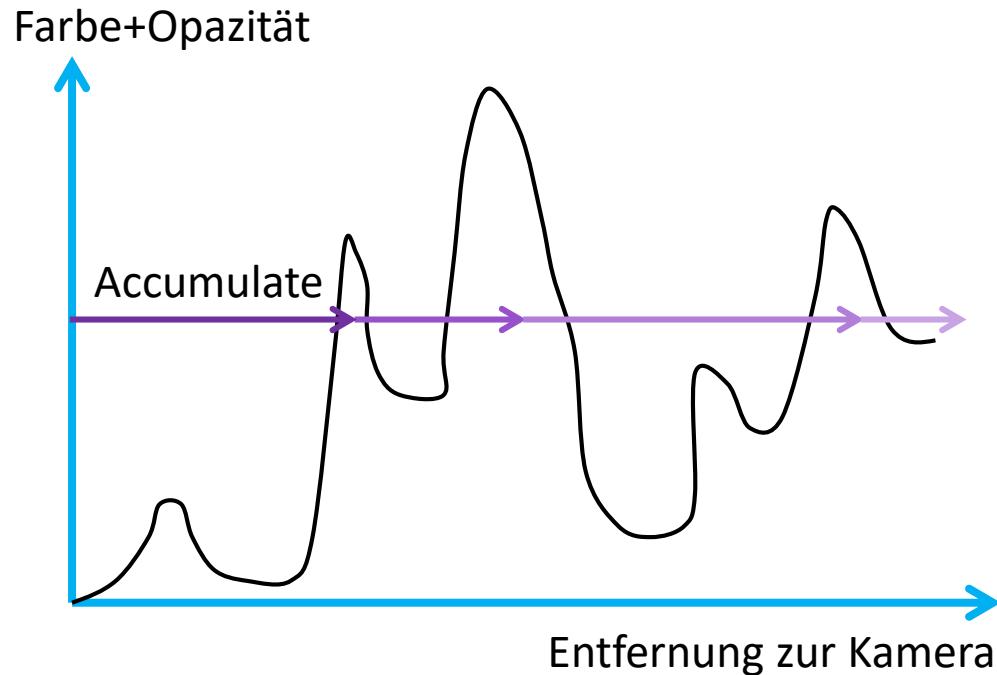
- ermittelt den höchsten Skalarwert für einen Sichtstrahls
- oft verwendet für Magnetresonanz und CT-Angiogramme
- gut geeignet um (Blut-)Gefäße darzustellen



Compositing

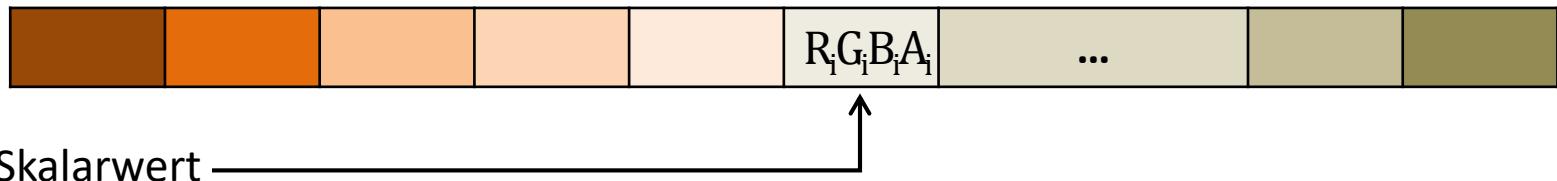
Compositing: Accumulate

- ▶ erlaubt die Darstellung (semi-)transparenter Bereiche
- ▶ basiert auf einem Emissions-Absorptionsmodell (Details später)
 - ▶ zugewiesene Farbe wird als emittiertes Licht betrachtet
 - ▶ semitransparente Bereiche lassen nur einen Teil des dahinter emittierten Lichts durch

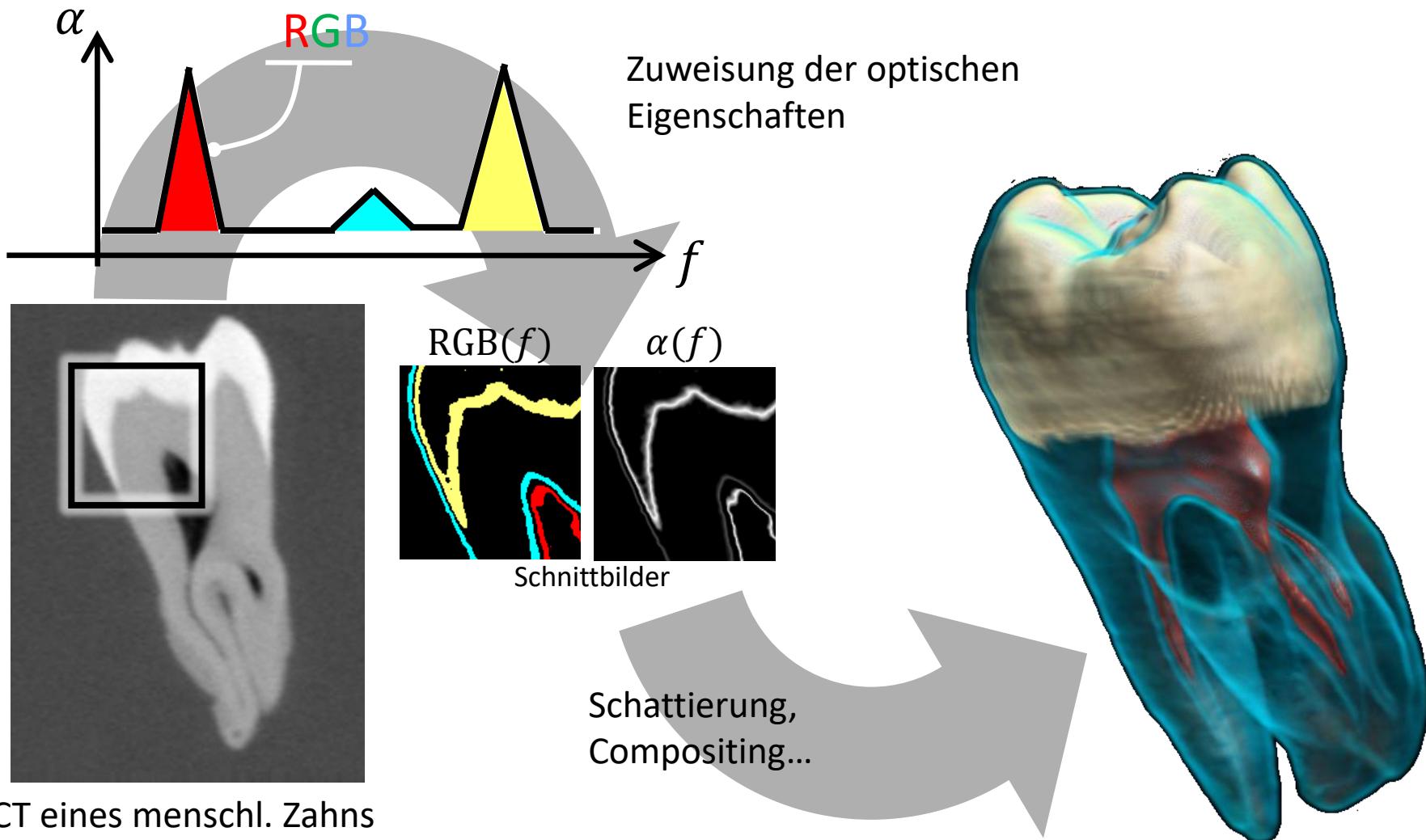


Transferfunktionen

- ▶ Ziel der Klassifikation ist die Unterscheidung von Arealen, wie z.B. Knochen, Haut, Muskeln, ...
- ▶ mit der Transferfunktion (TF) hofft man diesen Schritt zu Automatisieren
 - ▶ TF werden trotzdem oft interaktiv modifiziert oder nachgebessert
 - ▶ als Eingabe dienen Skalarwerte, Gradienten, Krümmung etc.
- ▶ Transferfunktion *im einfachsten Fall* („1D-Transferfunktion“)
 - ▶ jedem Skalarwert wird eine Farbe und Opazität zugewiesen
 - ▶ Transferfunktion T : Skalarwert \rightarrow RGBA
 - ▶ meist gespeichert in einer Lookup-Tabelle (1D Textur), die on-the-fly geändert werden kann



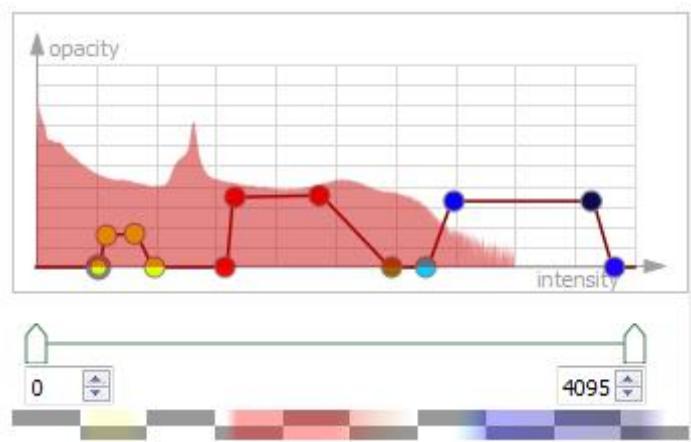
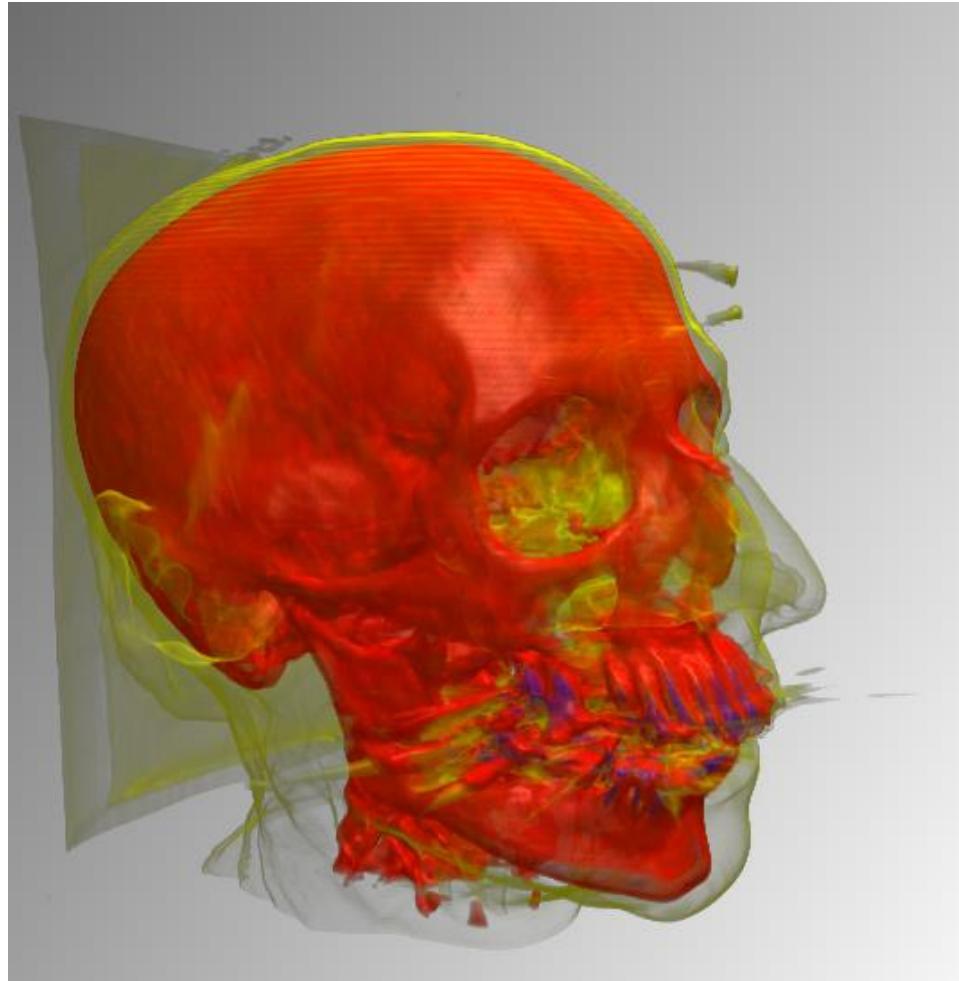
Transferfunktionen



Transferfunktionen

Beispiel

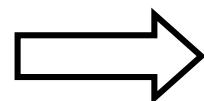
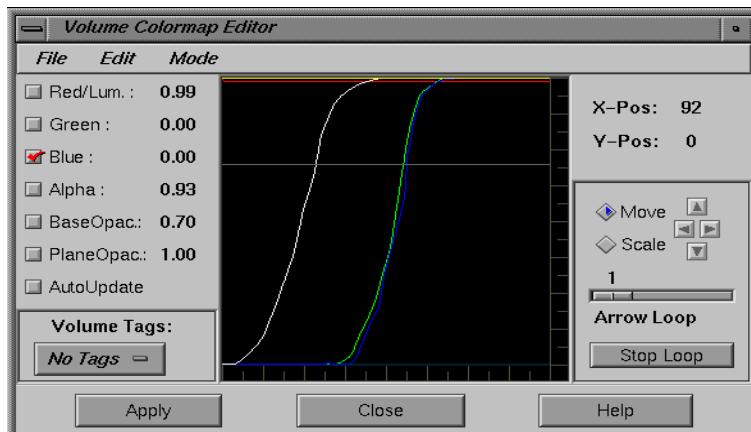
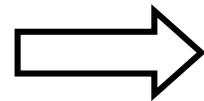
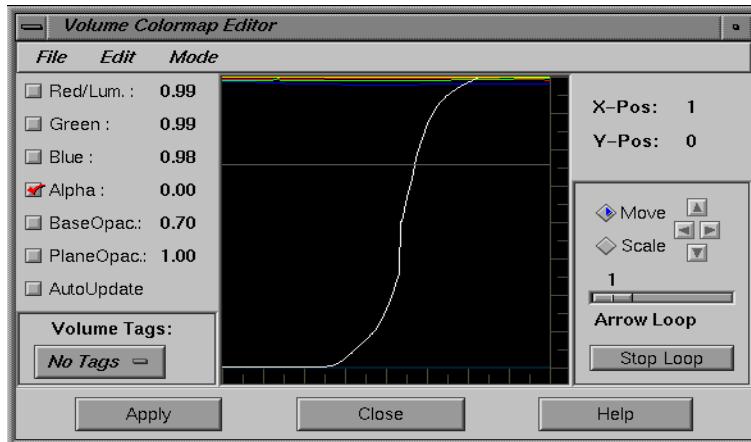
- Haut → semitransparentes Gelb, Knochen → (fast) opakes Rot etc.



Transferfunktionen

Beispiel

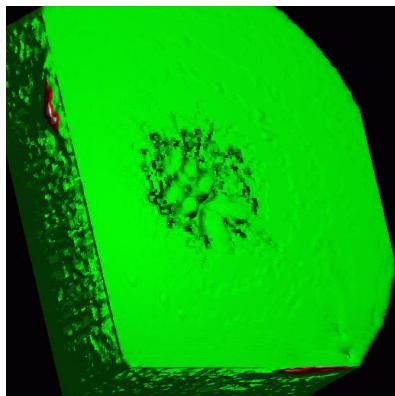
- hier wird lediglich die Opazität in Abhängigkeit vom Skalarwert festgelegt
- farbige Unterscheidung durch TF für rot, grün und blau



Transferfunktionen

Beispiel

- nur durch Änderung der TF erhält man eine Darstellung des umgebenden Mediums, der Schale und des weichen Gewebes

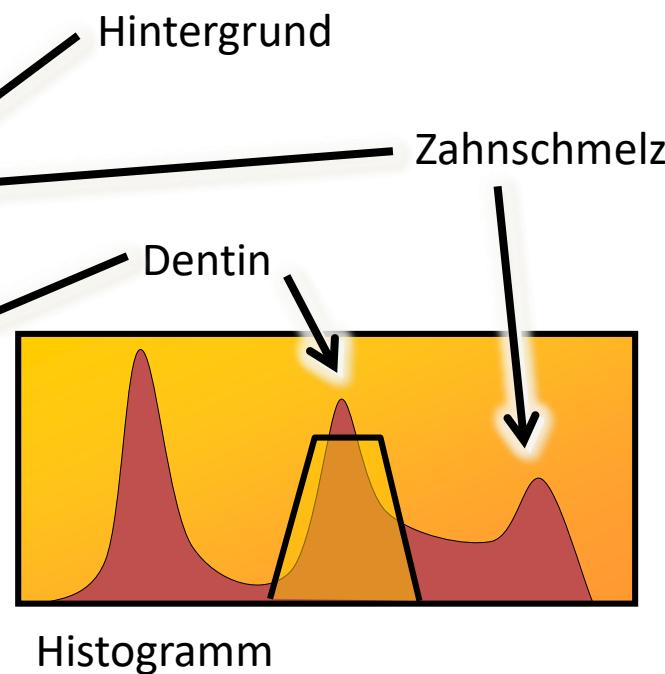
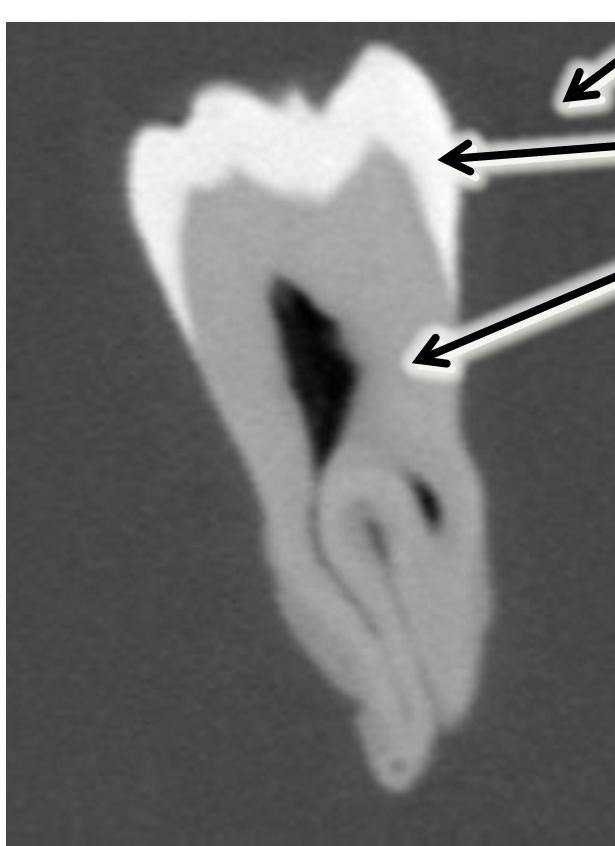


Transferfunktionen

1D-Transferfunktion (also ohne Berücksichtigung von Gradienten)

- ▶ durch Markierung der mittleren Skalarwerte wird der Zahnschmelz ebenfalls sichtbar

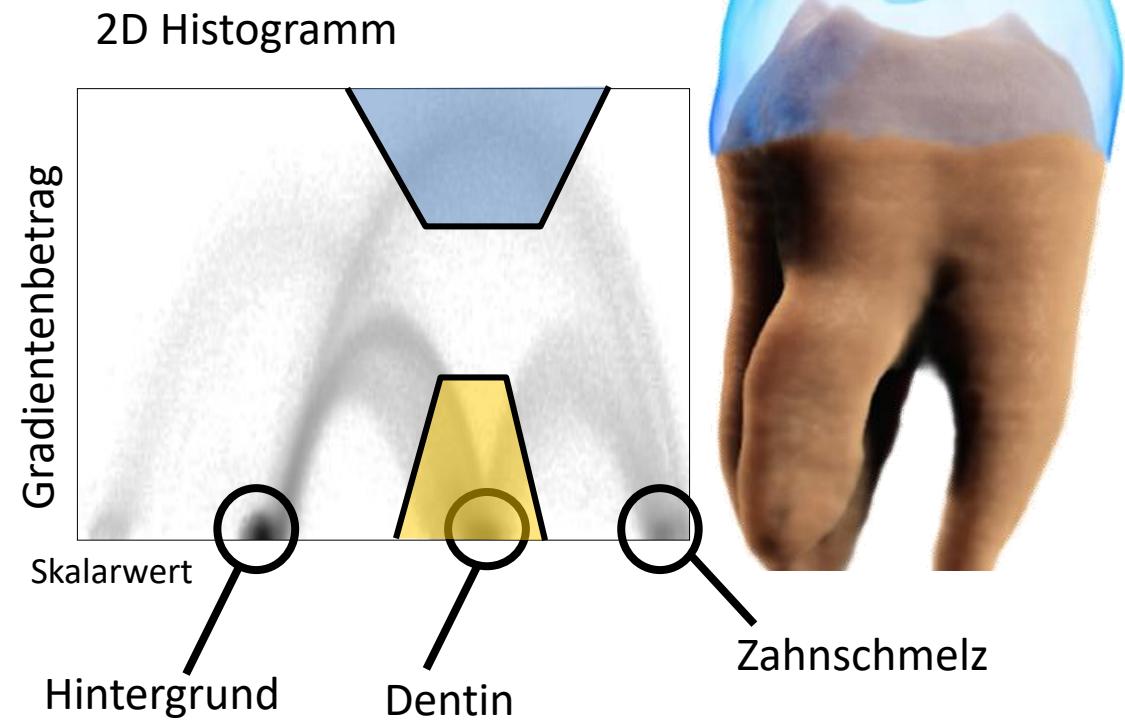
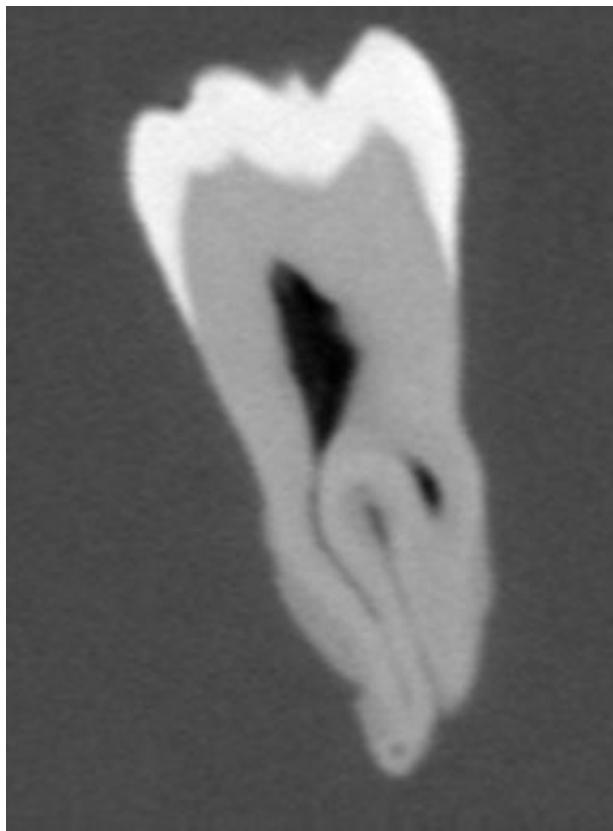
Schichtbild



Transferfunktionen

2D-Transferfunktion (mit Berücksichtigung des Gradienten)

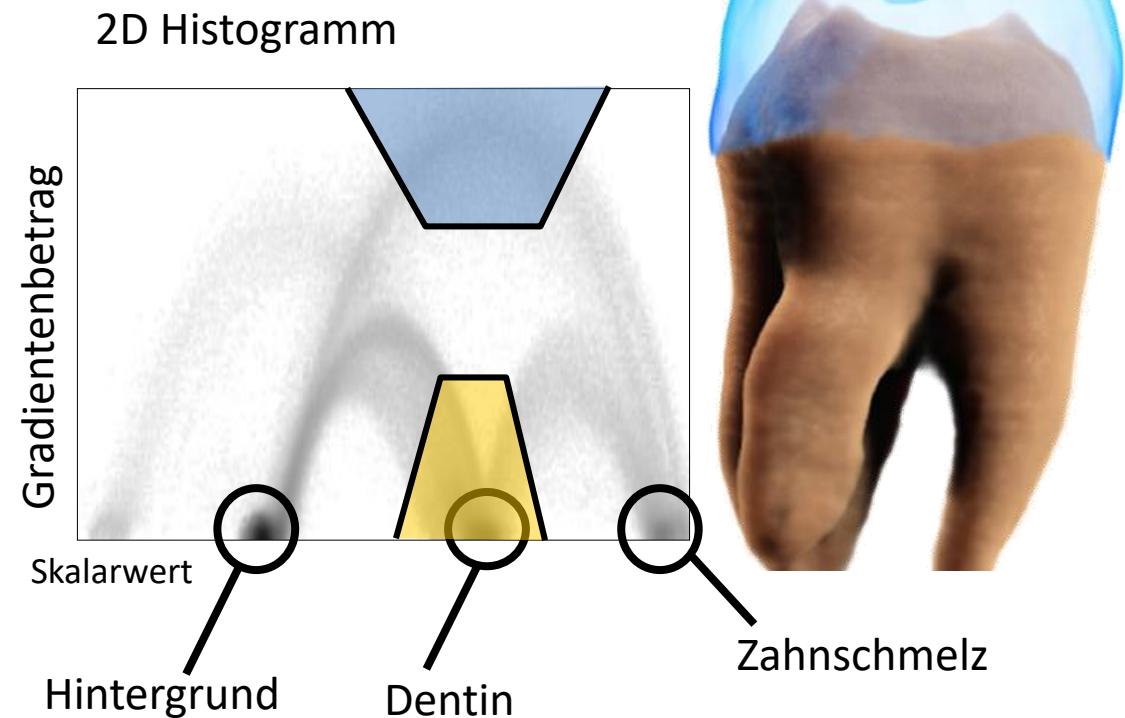
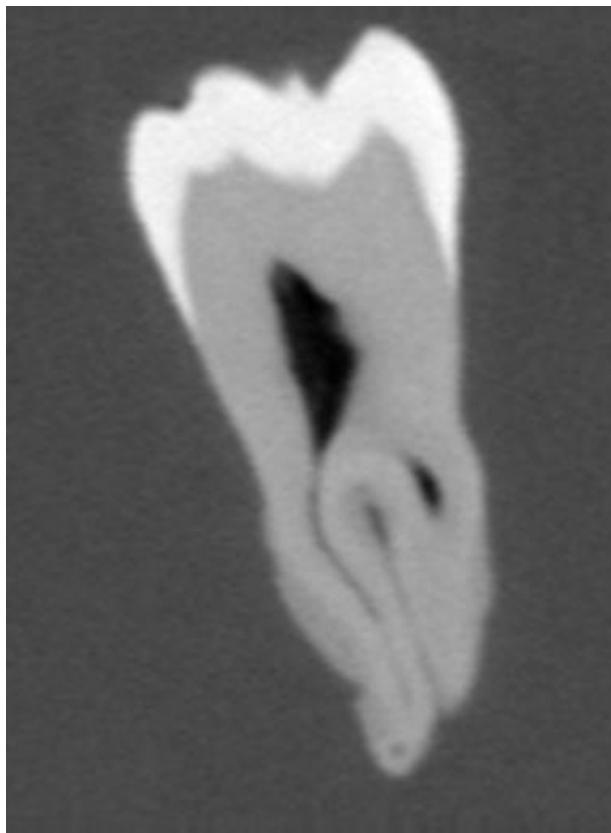
- ohne die Betrachtung des Gradienten wäre eine unterschiedliche Farbgebung für Zahnschmelz und Dentin nicht möglich



Transferfunktionen

2D-Transferfunktion (mit Berücksichtigung des Gradienten)

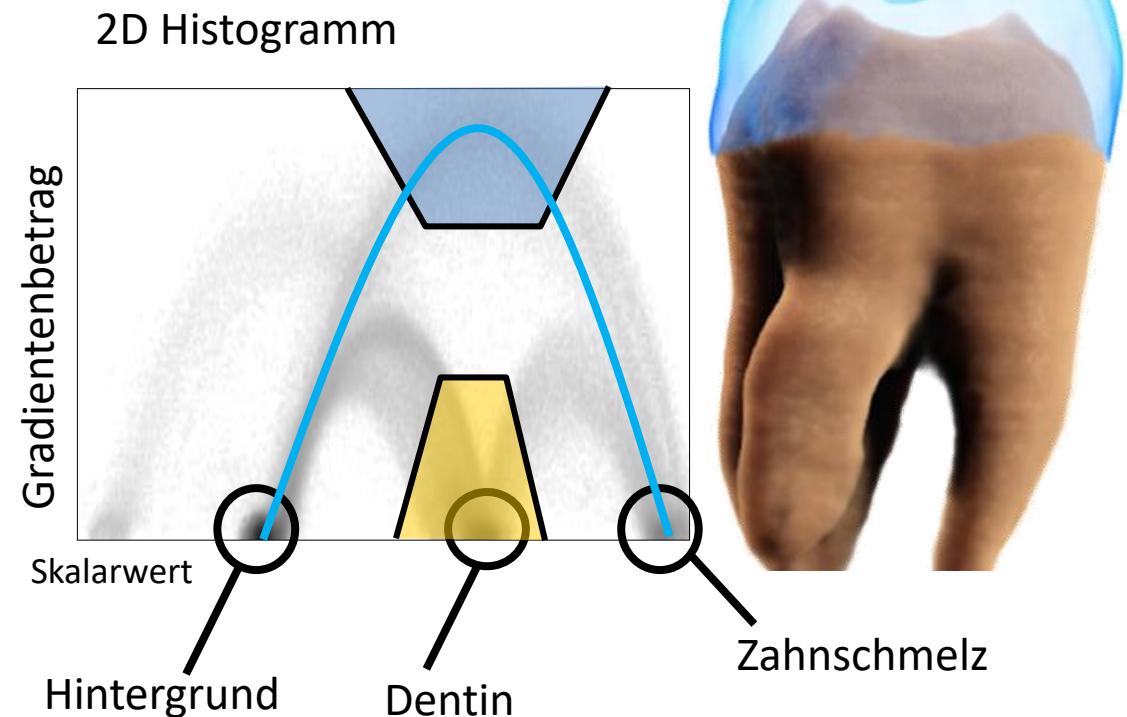
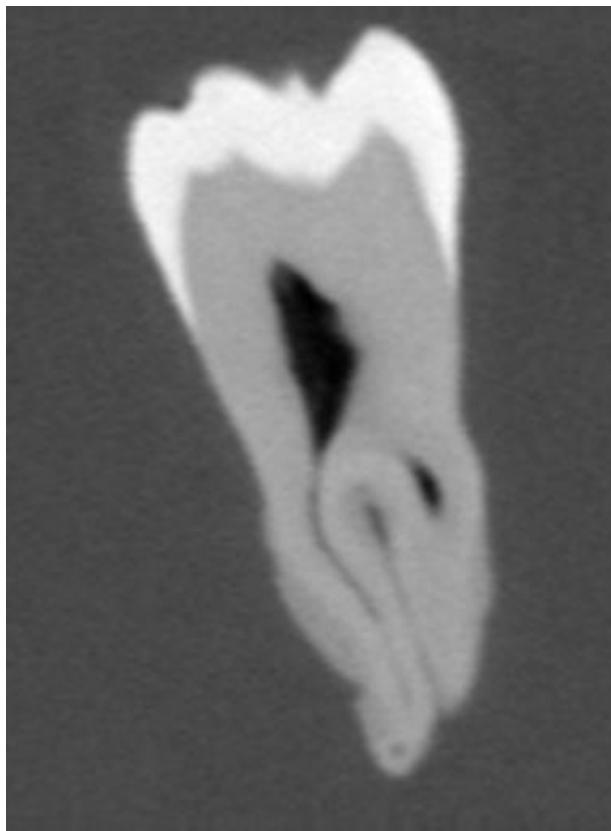
- dem gelben Bereich im Histogramm wird Farbe und Opazität zugewiesen: Skalarwerte die Dentin entsprechen und niedriger Gradient, d.h. nicht die Grenzschicht sondern der innere Bereich wird opak



Transferfunktionen

2D-Transferfunktion (mit Berücksichtigung des Gradienten)

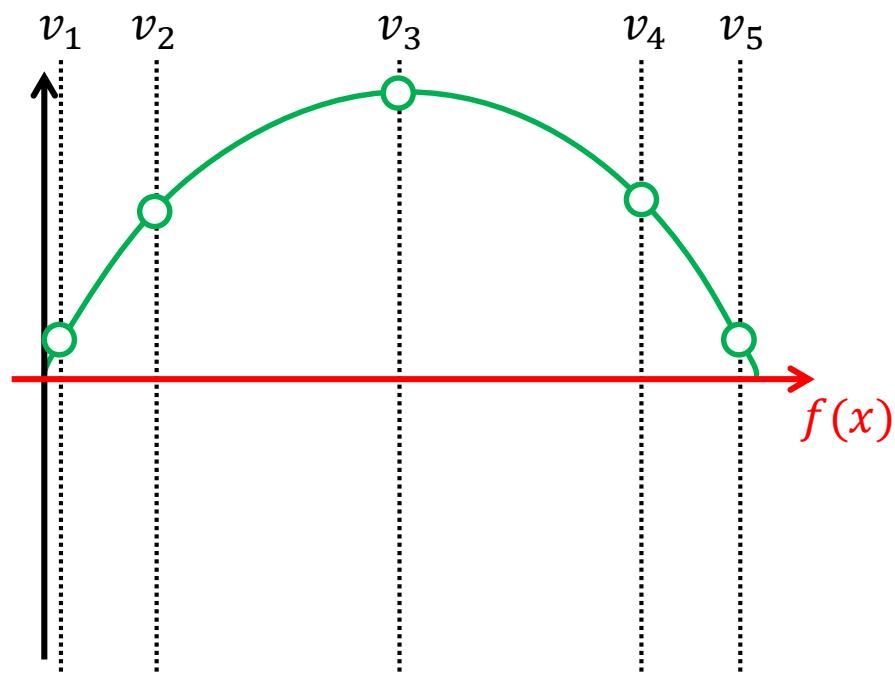
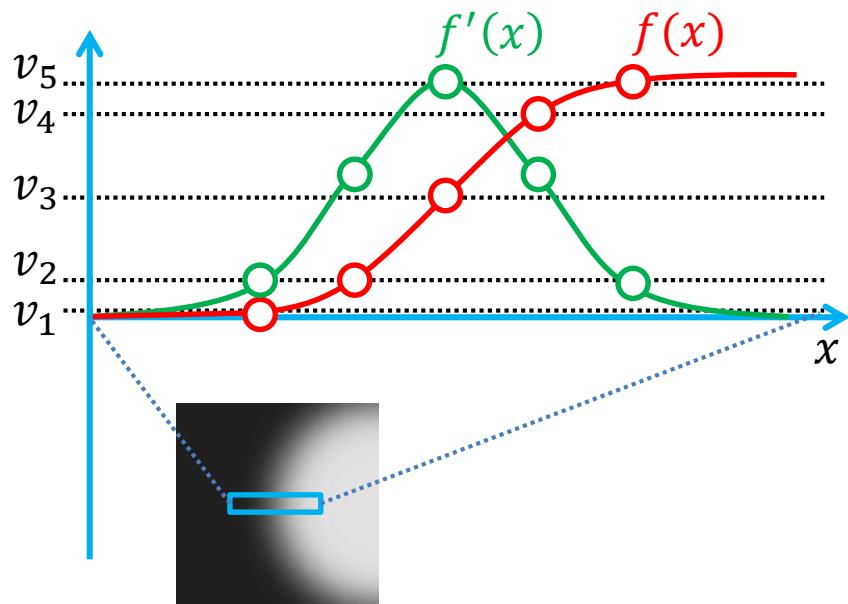
- Einträge im Histogramm entlang der Parabel entsprechen dem Übergang von Hintergrund zu Zahnschmelz: großer Unterschied im Skalarwert und daher hoher Gradient an der Grenzschicht



Transferfunktionen

2D-Transferfunktion

- ▶ woher kommt es, dass Voxel von Isoflächen „entlang von Bögen“ im Histogramm zu finden sind?



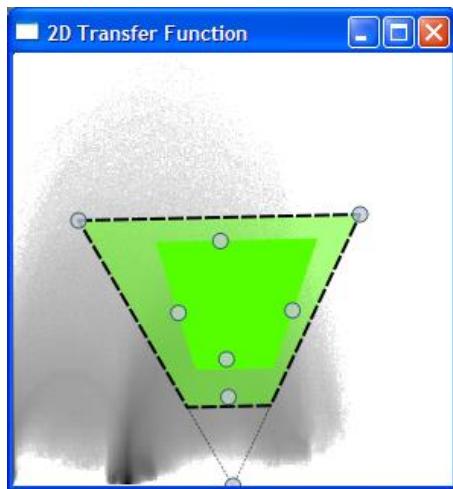
Transferfunktionen

Design von Transferfunktionen

- Editoren für 2D Transferfunktionen basieren meist auf (einfachen) geometrischen Primitiven
- Ziel: Freiheitsgrade reduzieren
- folgende Primitive haben sich in der Praxis bewährt

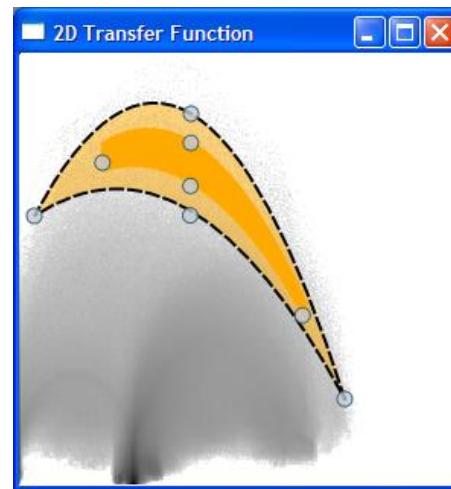
Trapezoide

(Kniss et al. 2001)

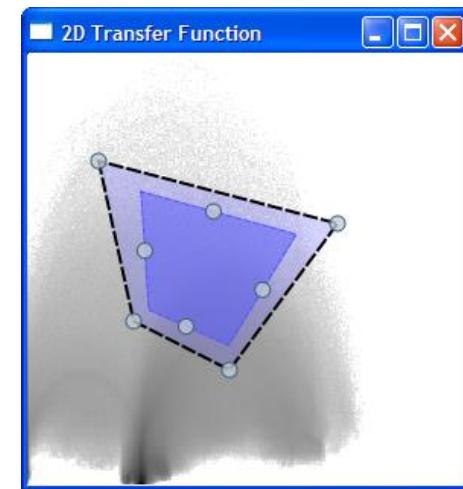


Paraboloide

(Vega et al. 2004)



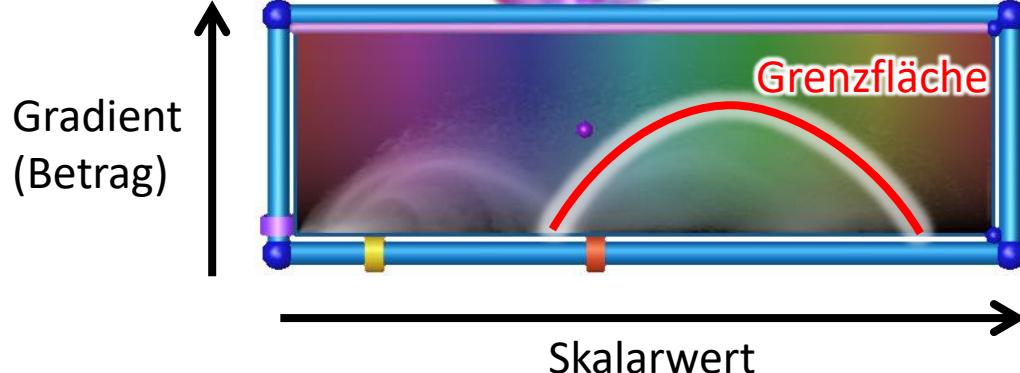
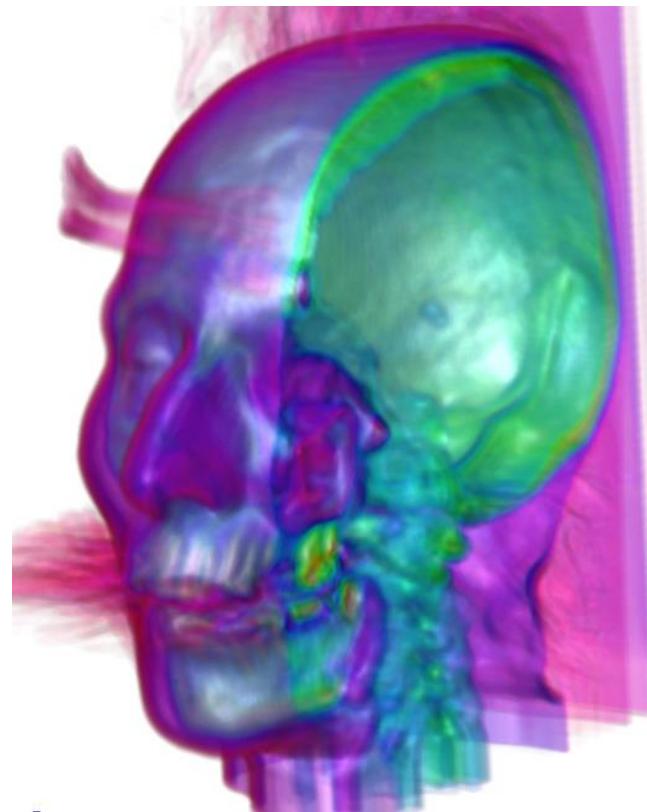
Vierecke



Transferfunktionen

Automatische Transferfunktionen

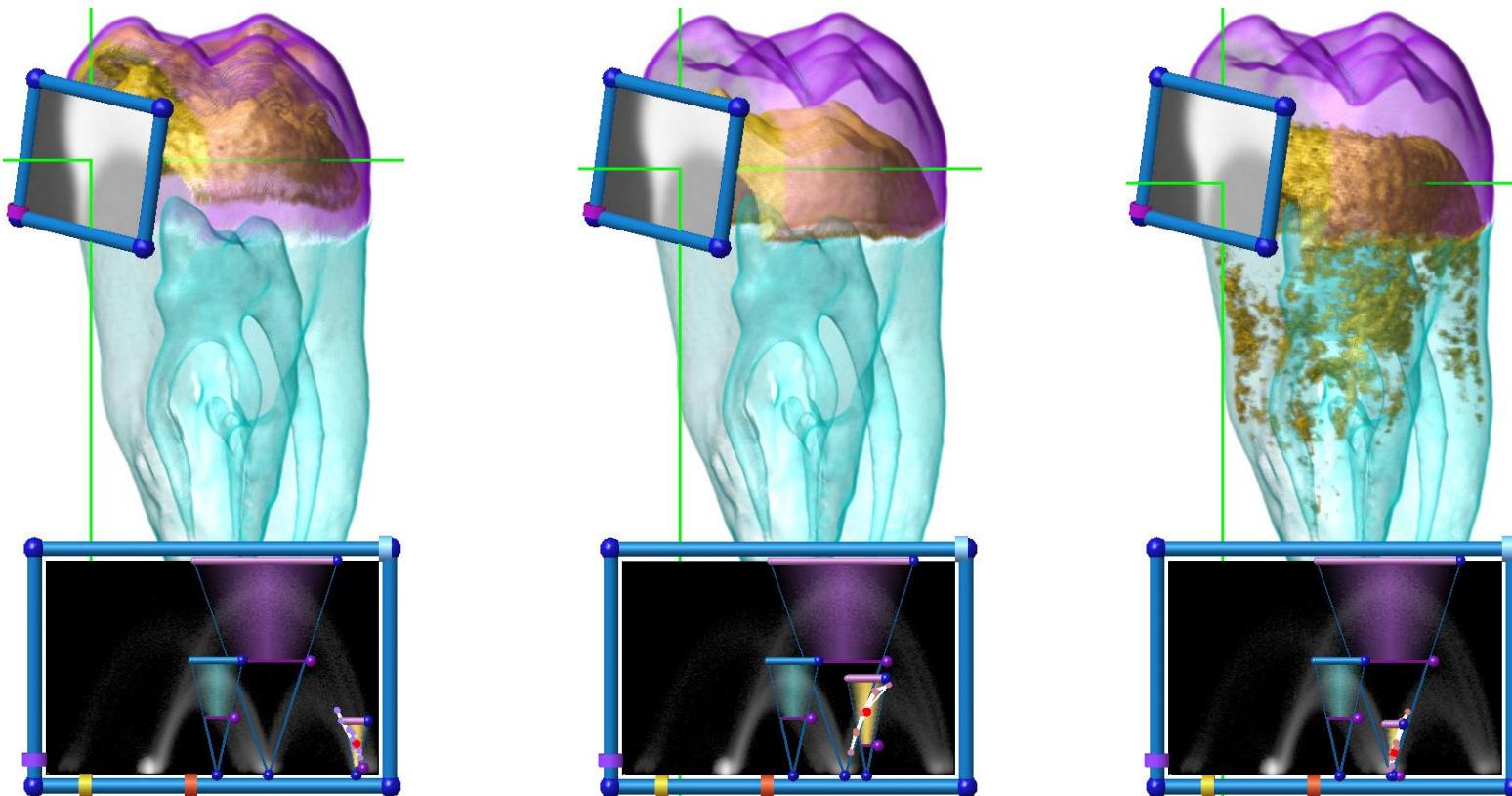
- ▶ bekommt man gute TF automatisch?
 - ▶ hier: Farbe variiert mit Skalarwert, Opazität mit Gradient
- ▶ in der Praxis:
 - ▶ Darstellung des Histogramms aus Skalarwert und Gradientenbetrag
 - ▶ interaktive Einstellung der Histogrammhelligkeiten (Skalierung der Gradienten, lineare vs. logarithmische Abbildung etc.)
 - ▶ manuelle Einstellung



Transferfunktionen

Unterstützung bei der Einstellung von Transferfunktionen

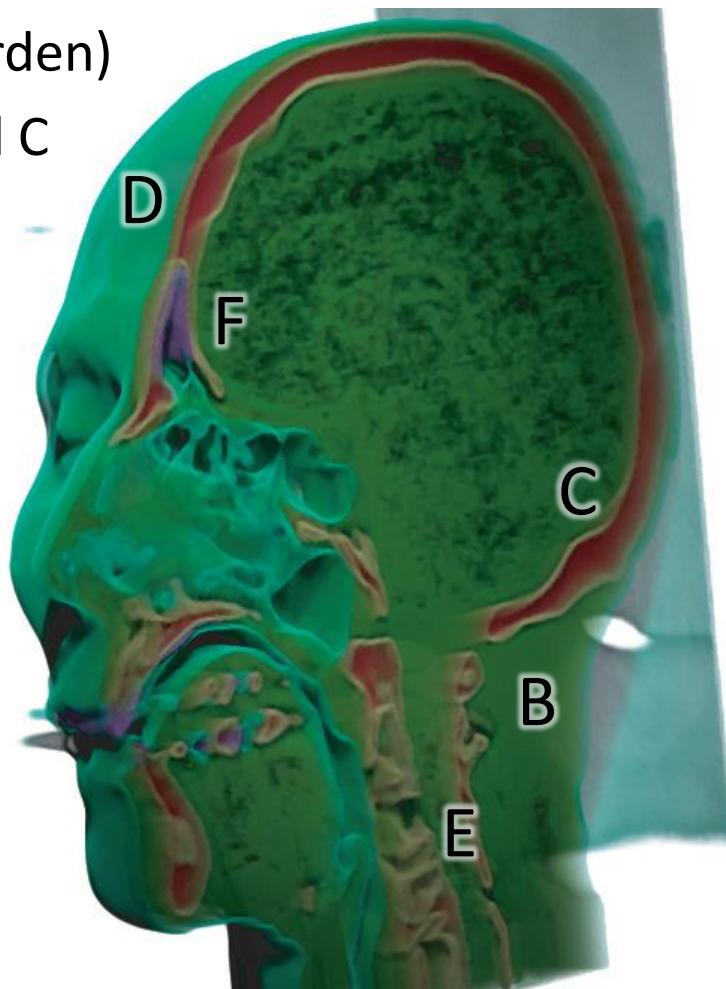
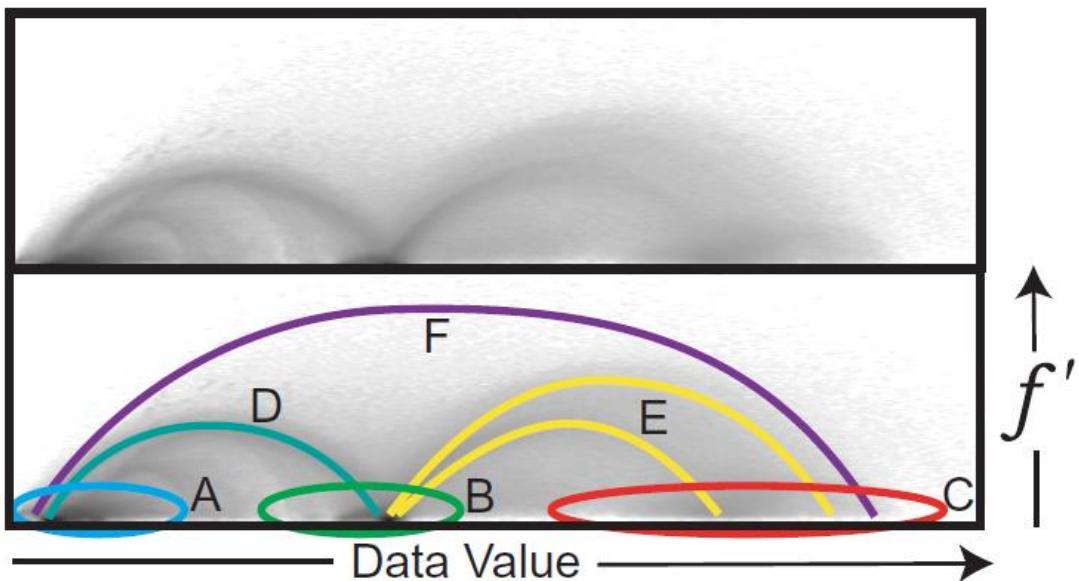
- ▶ mit einem Slicing-Window werden Bereiche des Volumens ausgewählt
- ▶ dort gefundene Datenwerte werden im Histogramm eingetragen, um die Einstellung der TF zu erleichtern



Transferfunktionen

Multidimensionale (2D) Transferfunktion

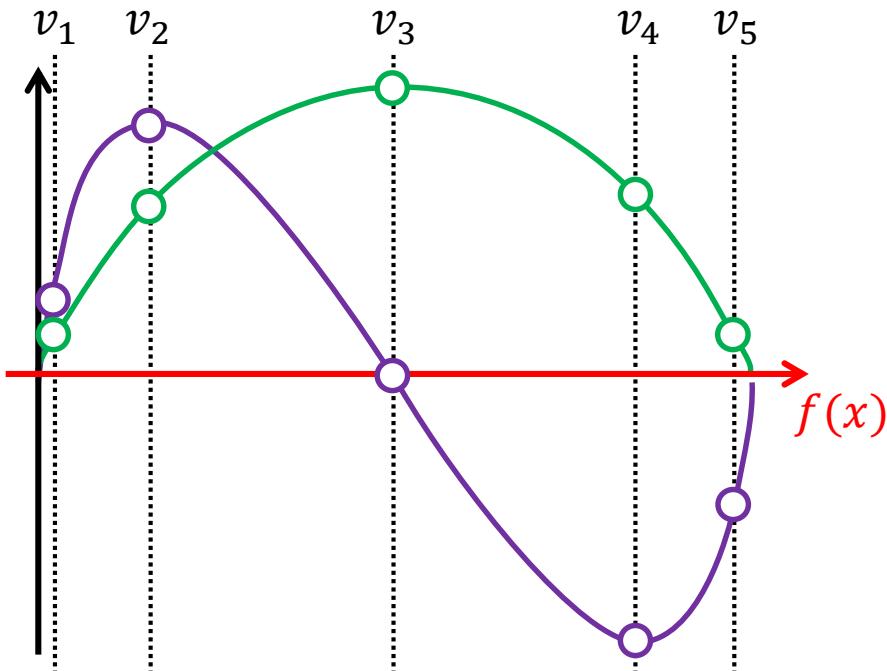
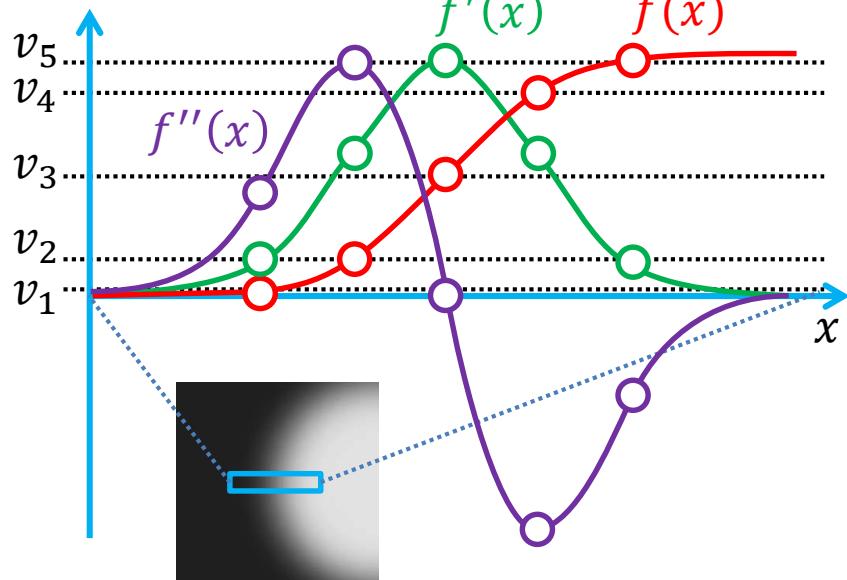
- ▶ A: Luft, B: weiches Gewebe, C: Knochen
(können auch durch eine 1D TF erfasst werden)
- ▶ D: Grenze zw. A und B, E: Grenze zw. B und C
(usw., benötigt 2D TF)



Transferfunktionen

Multidimensionale Transferfunktionen [Multi-Dimensional Transfer Functions for Interactive Volume Rendering, Kindlmann & Durkin 98, Kniss, Kindlmann, Hansen 01]

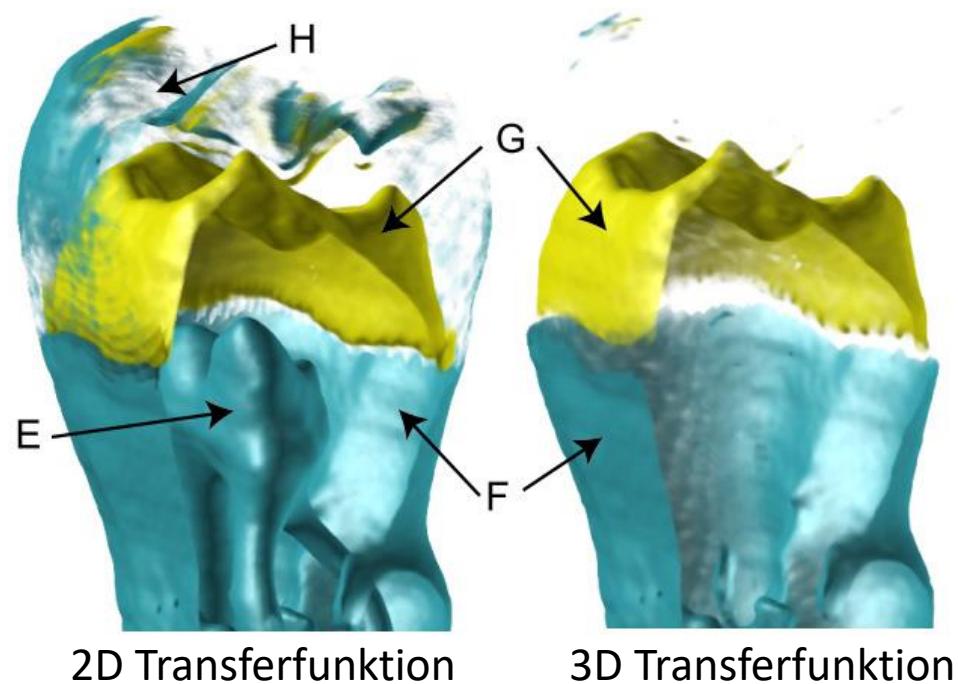
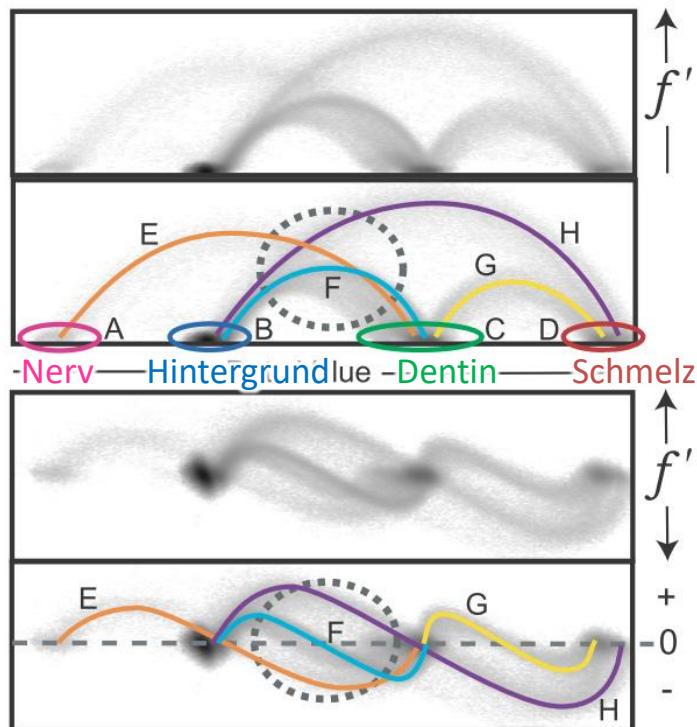
- ▶ Ziel (immer noch): Identifizieren von Grenzregionen/-flächen
- ▶ verwende 2D/3D Transferfunktionen abhängig von
 - ▶ Skalarwert, Betrag des Gradienten und
 - ▶ zweite Ableitung in Richtung des Gradienten
- ▶ nochmal zurück zu unserer Betrachtung der Ableitung(en): Steigung und Krümmung



Transferfunktionen

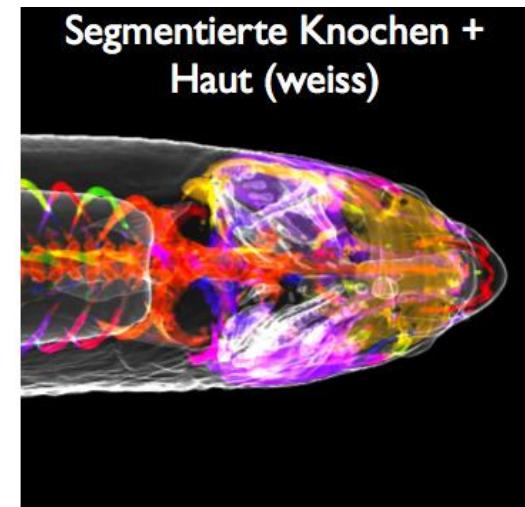
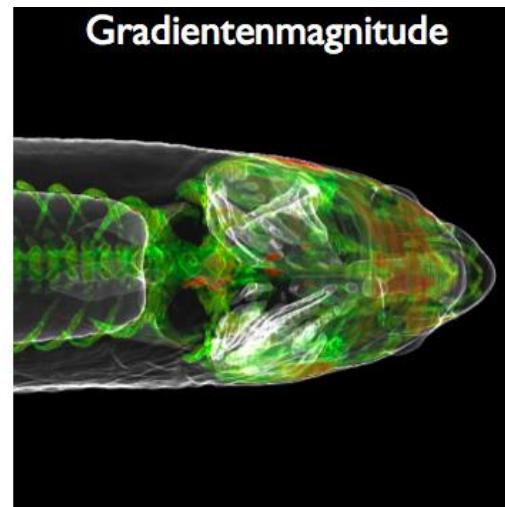
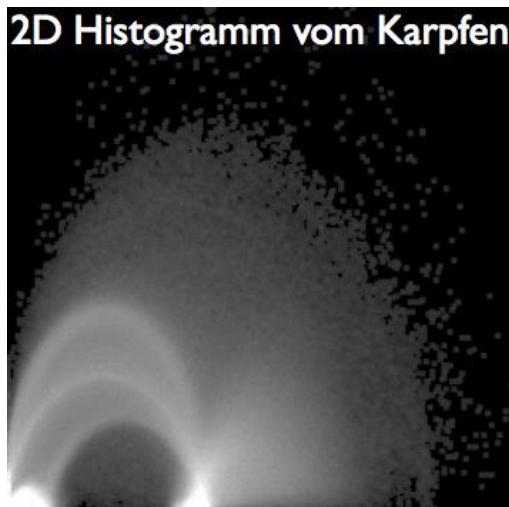
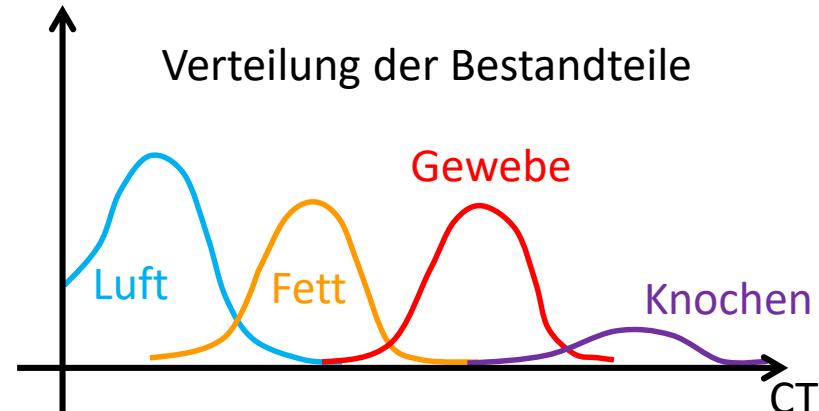
Multidimensionale (3D) Transferfunktion

- Ziel in dieser Darstellung: Hintergrund-Dentin F (gestrichelter Bereich in den Histogrammen) und Dentin-Zahnschmelz G Übergang
- 2D-TF: Nerv-Dentin (E) und Hintergrund-Zahnschmelz (H) werden fälschlicherweise ebenfalls opak(er)
- 3D-TF: reduziere Opazität dort, wo f'' von 0 abweicht



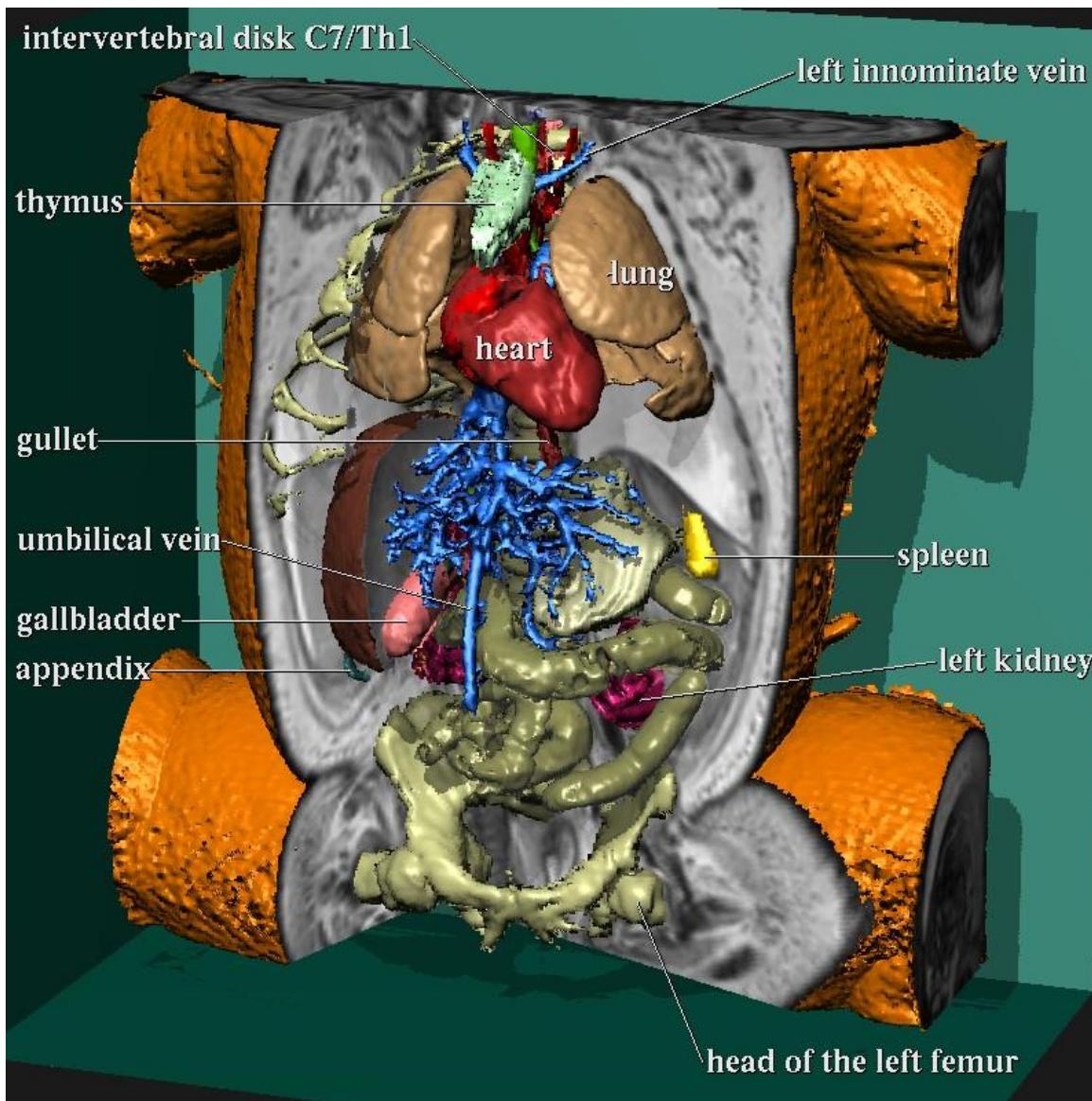
Segmentierung

- ▶ oft haben unterschiedliche Features ähnliche Skalarwerte, Gradienten, ...
 - ▶ z.B. bei CT: alle Organe absorbieren Röntgenstrahlung ähnlich
 - ▶ Klassifikation mittels TF genügt nicht zur Unterscheidung
 - ▶ besser: einfärben einzelner Knochen etc.
- ▶ Segmentierung ist ein (meist semi-automatischer) Vorverarbeitungsschritt zur Markierung der Voxel



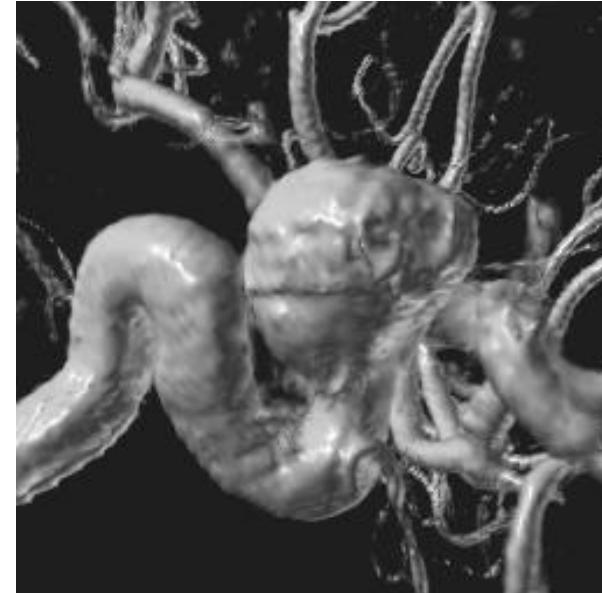
Segmentierung

Beispiel: Anatomischer Atlas



Unterstützung der Wahrnehmung der Form

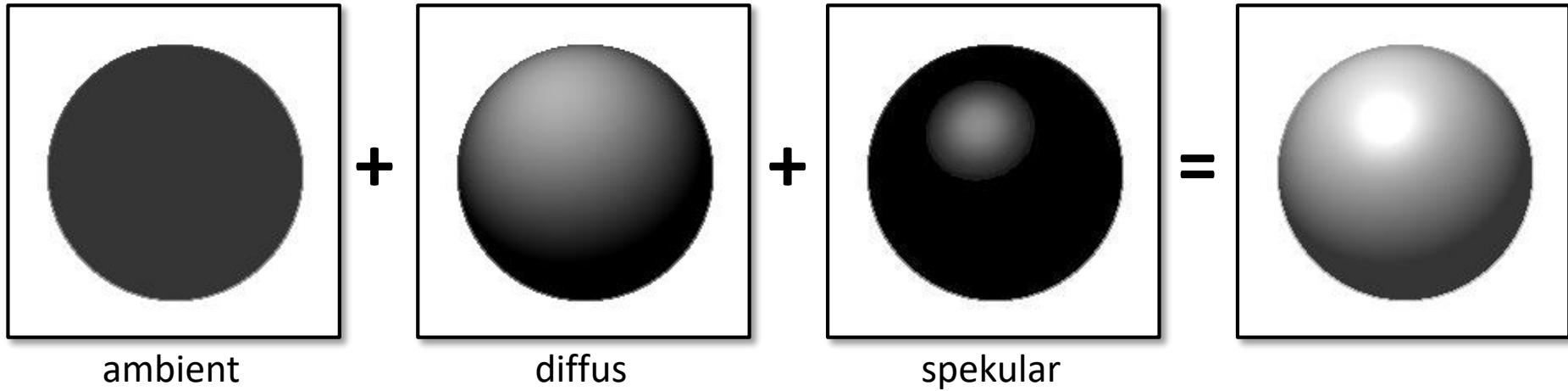
- ▶ „emittiertes“ Licht eines Voxels: Eigenfarbe (aus der Transferfunktion) ergänzt um / moduliert mit einem Reflexionsmodell
- ▶ lokale und globale Beleuchtung
 - ▶ wir betrachten zunächst nur lokale Beleuchtung, also reflektiertes Licht (und Absorption in semitransparenten Volumen)
 - ▶ wird (Mehrfach-)Streuung im Volumen berücksichtigt, spricht man von globaler Beleuchtung (siehe Vorlesung Fotorealistische Bildsynthese)



Phong Beleuchtungsmodell

- ▶ phänomenologisches Modell, drei Komponenten
 - ▶ ambient I_a , diffus I_d und spekular I_s : $I = I_a + I_d + I_s$

$$I = k_a \cdot I_L + k_d \cdot I_L \cdot (\mathbf{N} \cdot \mathbf{L}) + k_s \cdot I_L \cdot (\mathbf{R}_L \cdot \mathbf{V})^n$$



- ▶ Anmerkung: man kann natürlich jede beliebige BRDF verwenden

Phong Beleuchtungsmodell

- phänomenologisches Modell, drei Komponenten
 - ambient I_a , diffus I_d und spekular I_s : $I = I_a + I_d + I_s$

$$I = k_a \cdot I_L + k_d \cdot I_L \cdot (\mathbf{N} \cdot \mathbf{L}) + k_s \cdot I_L \cdot (\mathbf{R}_L \cdot \mathbf{V})^n$$

- Stärke der Spiegelung fällt für von \mathbf{R}_L unterschiedl. Richtungen stark ab
 - modelliere Abfall durch $\cos^n \alpha$ mit „Phong-Exponent“ n

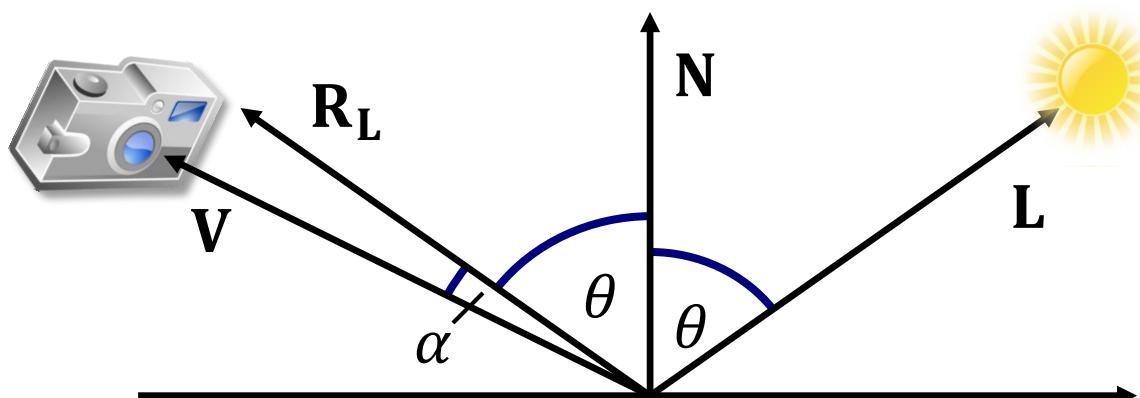
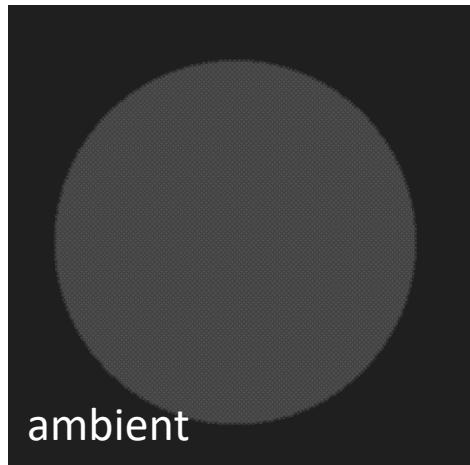
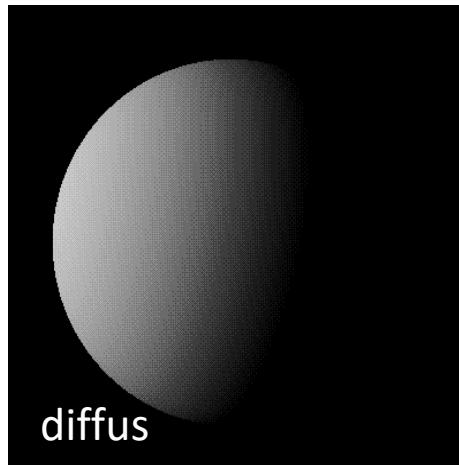


Bild: Andrea Weidlich

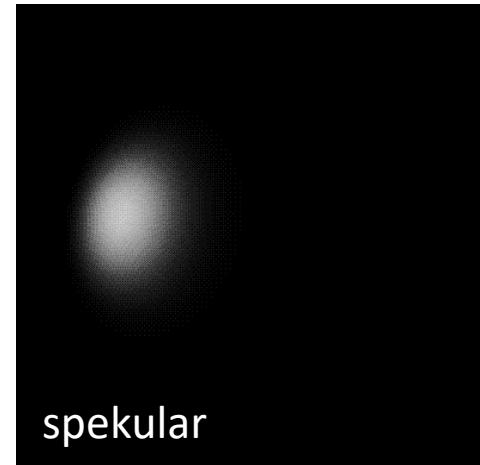
Lokale Beleuchtung (Phong)



ambient

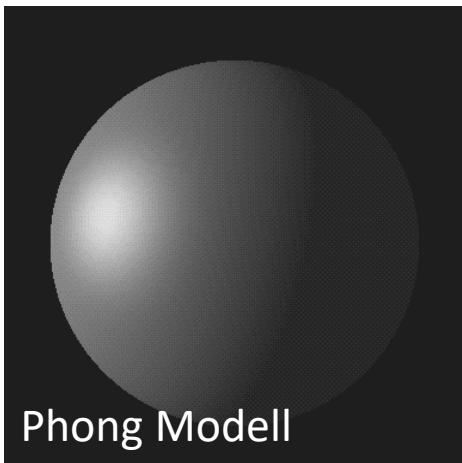


diffus



spekular

$$\begin{aligned} k_a &= 0.1 \\ k_d &= 0.5 \\ k_s &= 0.4 \end{aligned}$$



Phong Modell

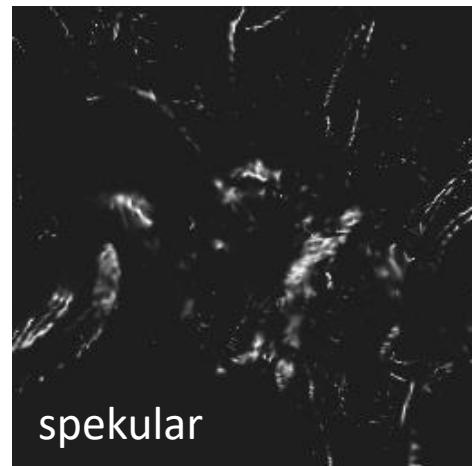
Lokale Beleuchtung (Phong)



ambient

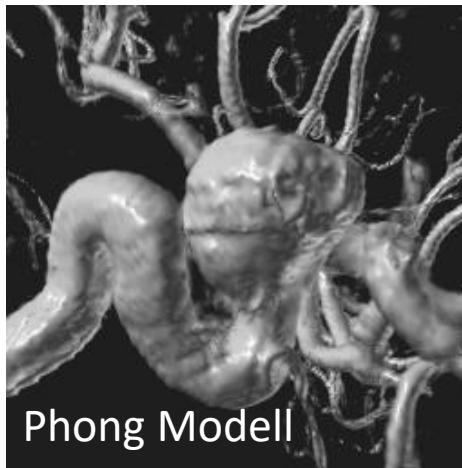


diffus



spekular

$$\begin{aligned} k_a &= 0.1 \\ k_d &= 0.5 \\ k_s &= 0.4 \end{aligned}$$



Phong Modell

- was ist der Normalenvektor eines Skalarfelds?
- der Gradientenvektor $\mathbf{grad} f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right)^T = \nabla f$ steht senkrecht auf den Isoflächen und zeigt in Richtung der größten Veränderung
- der Gradientenvektor ersetzt daher die Normale bei der Beleuchtungsberechnung (auch in semi-transparenten Bereichen in denen eigentlich keine richtige Fläche zu erkennen ist)
- aber Vorsicht in Bereichen mit zu kleinem Gradientenbetrag:
Rauschen in den Daten kann zu stark variierenden Richtungen führen
- numerische Berechnung des Gradienten
 - zentrale Differenzen
 - Vorwärts-/Rückwärtsdifferenzen
 - Sobel Operator (3×3 Kernel für jede partielle Ableitung)

Lokale Beleuchtung

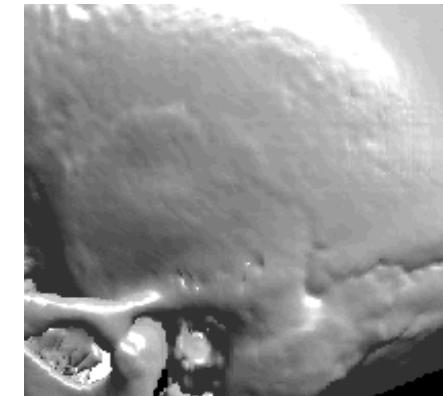
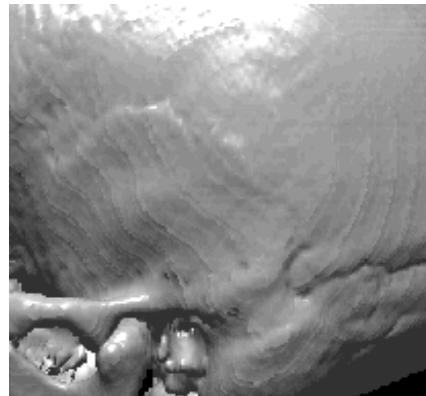
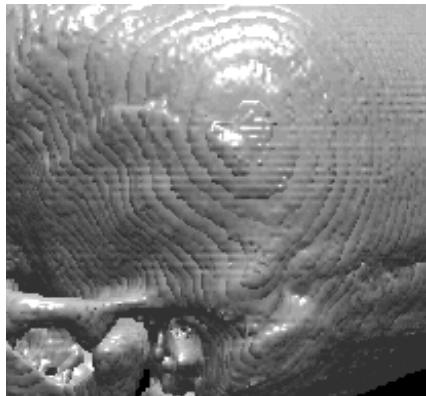
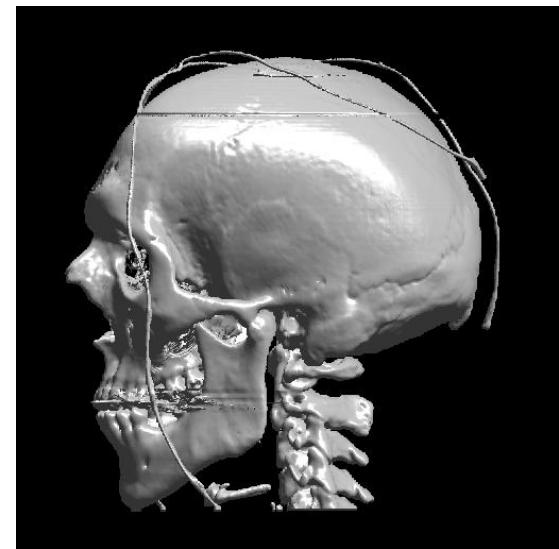
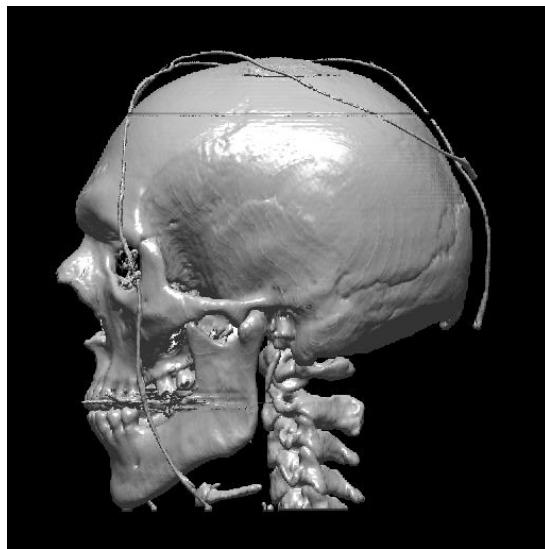
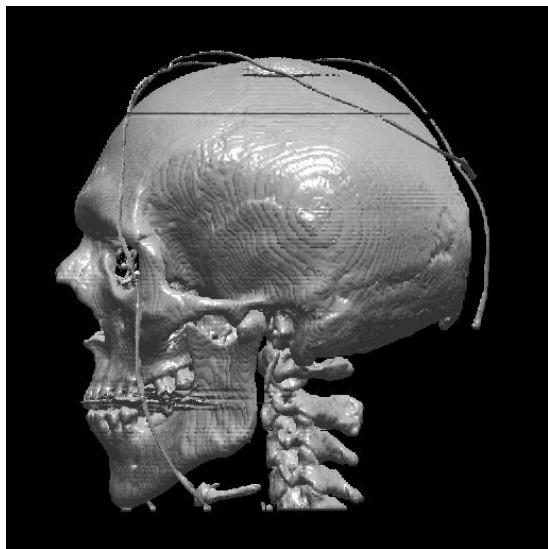


Vorwärts-/Rückwärtsdifferenzen



zentrale Differenzen

Lokale Beleuchtung



Vorwärts-/Rückwärtsdifferenzen

zentrale Differenzen

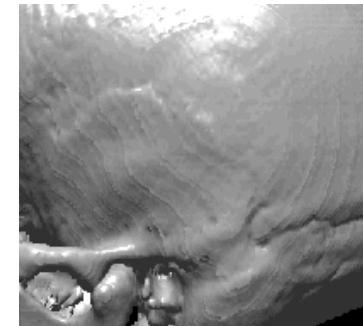
Sobel Operator

Gradient in einem 3D-Gitter mit zentralen Differenzen

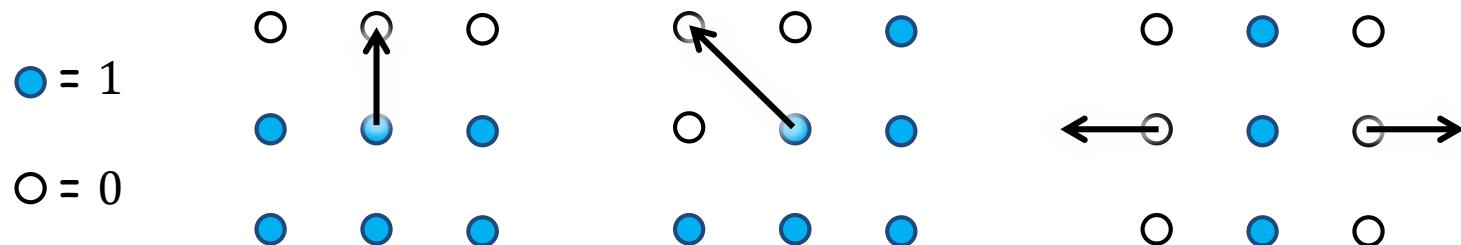
► Berechnung $G_x = V_{x+1,y,z} - V_{x-1,y,z}$

$$G_y = V_{x,y+1,z} - V_{x,y-1,z}$$

$$G_z = V_{x,y,z+1} - V_{x,y,z-1}$$



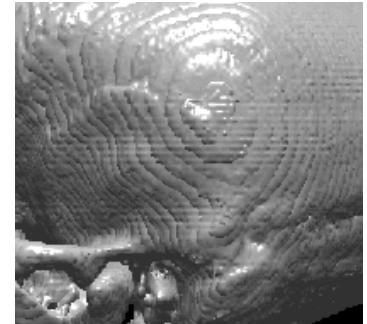
- entspricht Faltung mit dem Kernel $[-1 \ 0 \ 1]$ (Hochpassfilter)
- auf diese Weise berechnete Gradienten sind nicht isotrop
 - je nach Orientierung der Isofläche ist die Länge zwischen 1 und $\sqrt{3}$
 - Normalisierung ist notwendig
 - Beispiel:



Lokale Beleuchtung

► Vorwärts-/Rückwärtsdifferenzen

- ▶ Faltungs-Kernel $[-1 \ 1]$ (ebenfalls ein Hochpassfilter)
- ▶ sehr empfindlich gegenüber verrauschten Daten
- ▶ nicht isotrop

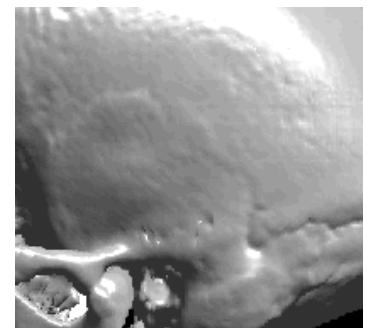


► Sobel Operator

- ▶ beinahe isotrop
- ▶ teurere Berechnung (mehrere Multiplikationen und Additionen)
- ▶ berechnet Ableitung und glättet gleichzeitig orthogonal zur Ableitungsrichtung
- ▶ partielle Ableitung z -Achse (andere Richtungen durch Rotation): Glättung in x - und y -Richtung, Differenz in z -Richtung

$$\begin{bmatrix} -1 & -3 & -1 \\ -3 & -6 & -3 \\ -1 & -3 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 3 & 1 \\ 3 & 6 & 3 \\ 1 & 3 & 1 \end{bmatrix}$$

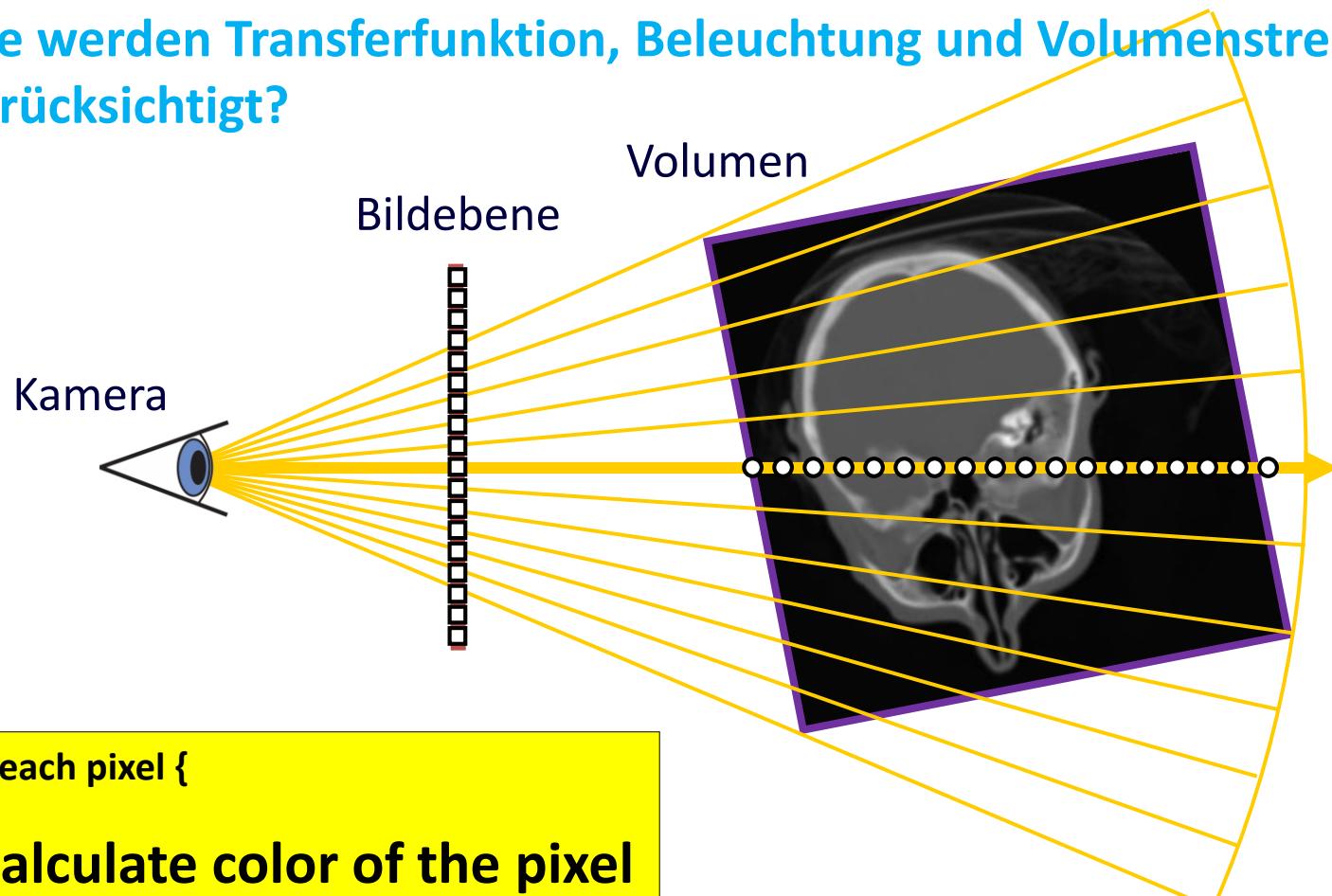
Kernel für Kernel für nächstes
vorheriges Slice aktuelles z -Slice Slice



Direkte Volumenvisualisierung

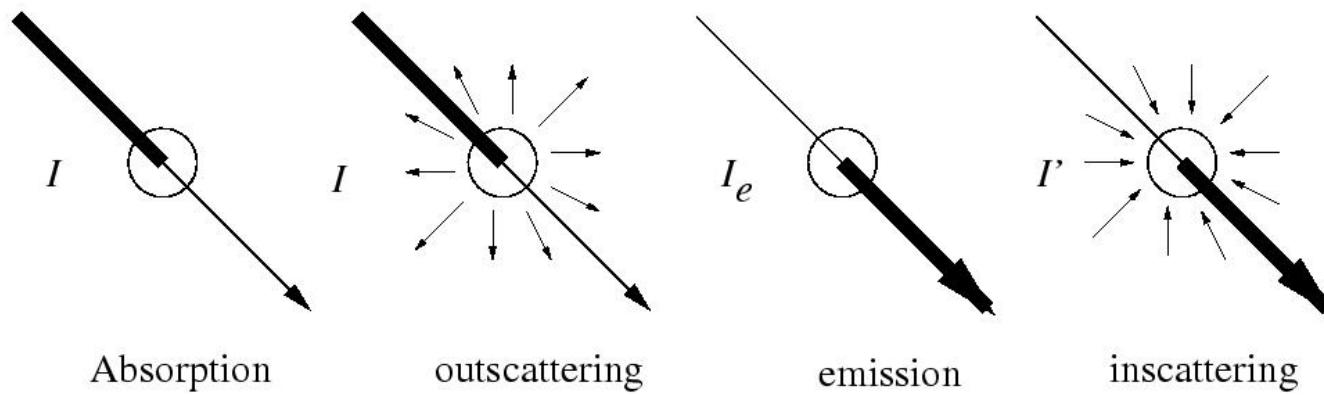
Image-Order Ansatz: Raycasting / Raymarching

- wie werden Transferfunktion, Beleuchtung und Volumenstreuung berücksichtigt?



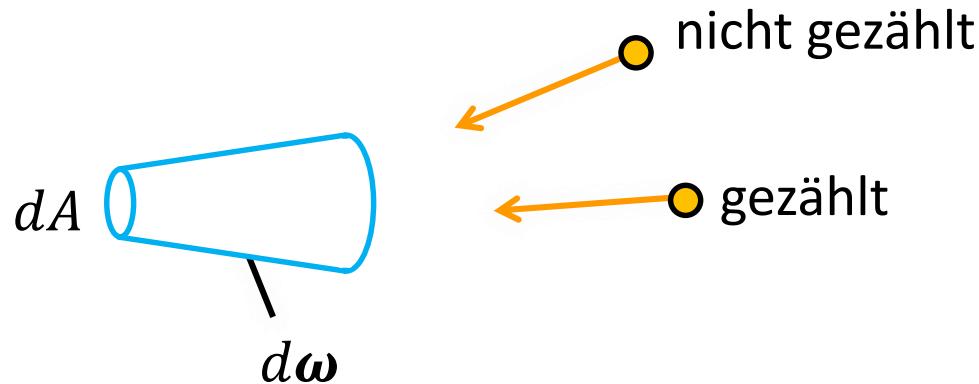
Volumen Rendering Gleichung

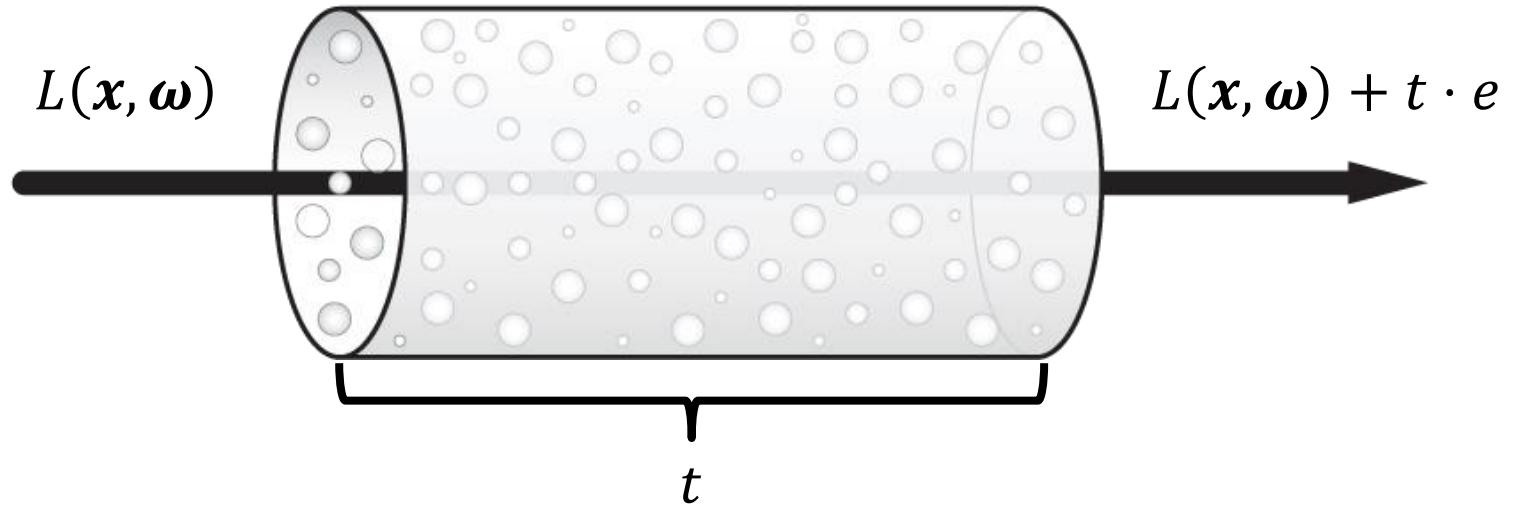
- Ziel: physikalisches (oder zumindest physikalisch-motiviertes) Modell für Volumenrendering
 - in der Praxis verwendet man meist vereinfachte Modelle, z.B. ein Emissions-Absorptions-Modell, das die Betrachtung auf einen Sichtstrahl beschränkt
- wichtige Aspekte für eine volumetrische Darstellung
 - Absorption (\rightarrow Zusammenhang mit Transparenz/Opazität)
 - Emission (und Beleuchtung)
 - Streuung



Strahldichte (Radiance)

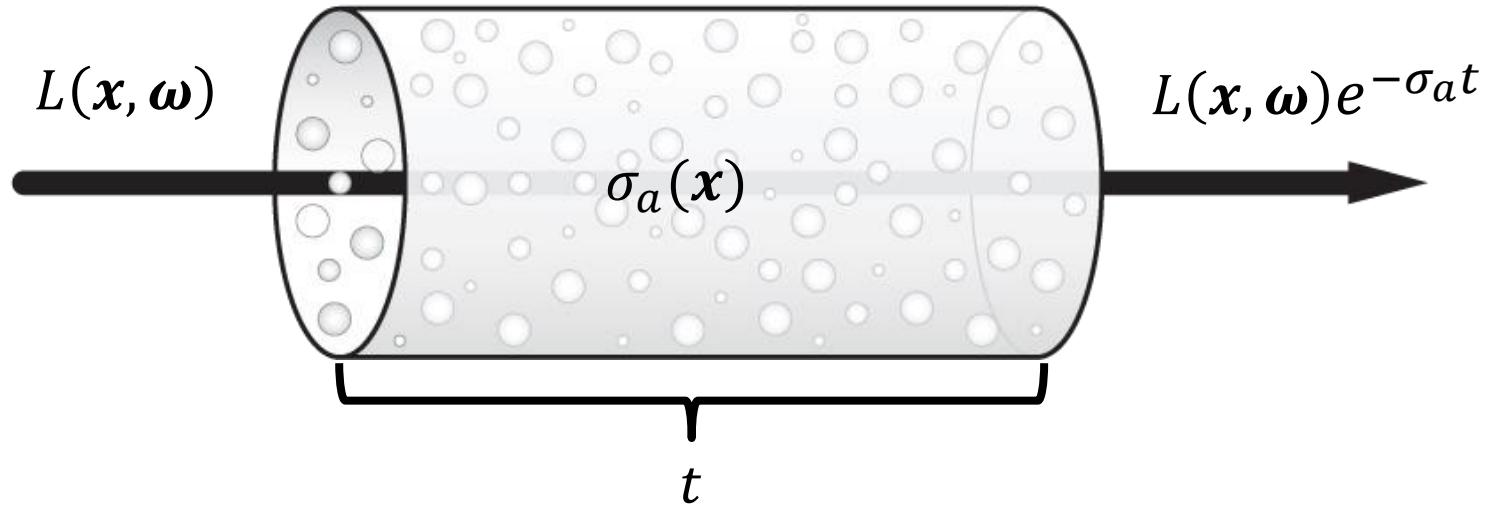
- ▶ ...physikalische Größe, die beschreibt wie viel „Licht“ entlang eines Strahls (also in einer bestimmten Richtung) transportiert wird
 - ▶ vergleichbar mit dem, was unser Auge misst
 - ▶ Radiance $L = \frac{d\Phi}{dAd\omega} = \frac{dQ}{dAd\omega dt}$ in $\left[\frac{\text{W}}{\text{m}^2\text{sr}} \right]$:
Energie Q (von Photonen) pro Fläche dA , Raumwinkel $d\omega$ und Zeit dt
- ▶ vereinfachte Vorstellung: „Messapparat“, der die Energie der auftreffenden Photonen durch einen kegelförmigen Trichter misst





- ▶ Emission ist additiv, die Radiance entlang des Strahls nimmt zu
- ▶ entlang eines Strahls ist die Emission entweder...
 - ▶ ...konstant
$$L(t) := L(x + t\omega, \omega) = L(x, \omega) + t \cdot e$$
 - ▶ oder nicht konstant
$$\frac{d}{dt} L(t) = e(t)$$

→ Gesamtemission durch Integration

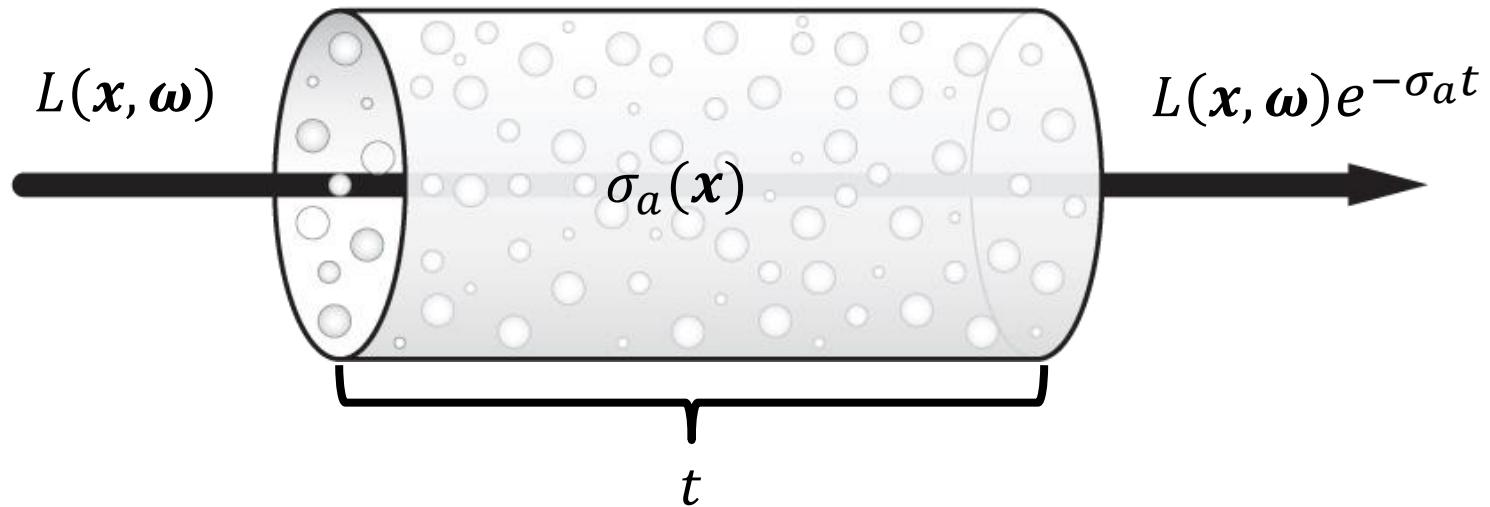


- Absorption: „Wahrscheinlichkeit, dass ein Photon pro Längeneinheit absorbiert wird“ → exponentielle Abnahme der Radiance

$$L(x + t\omega, \omega) = L(x, \omega)e^{-\sigma_a t}$$

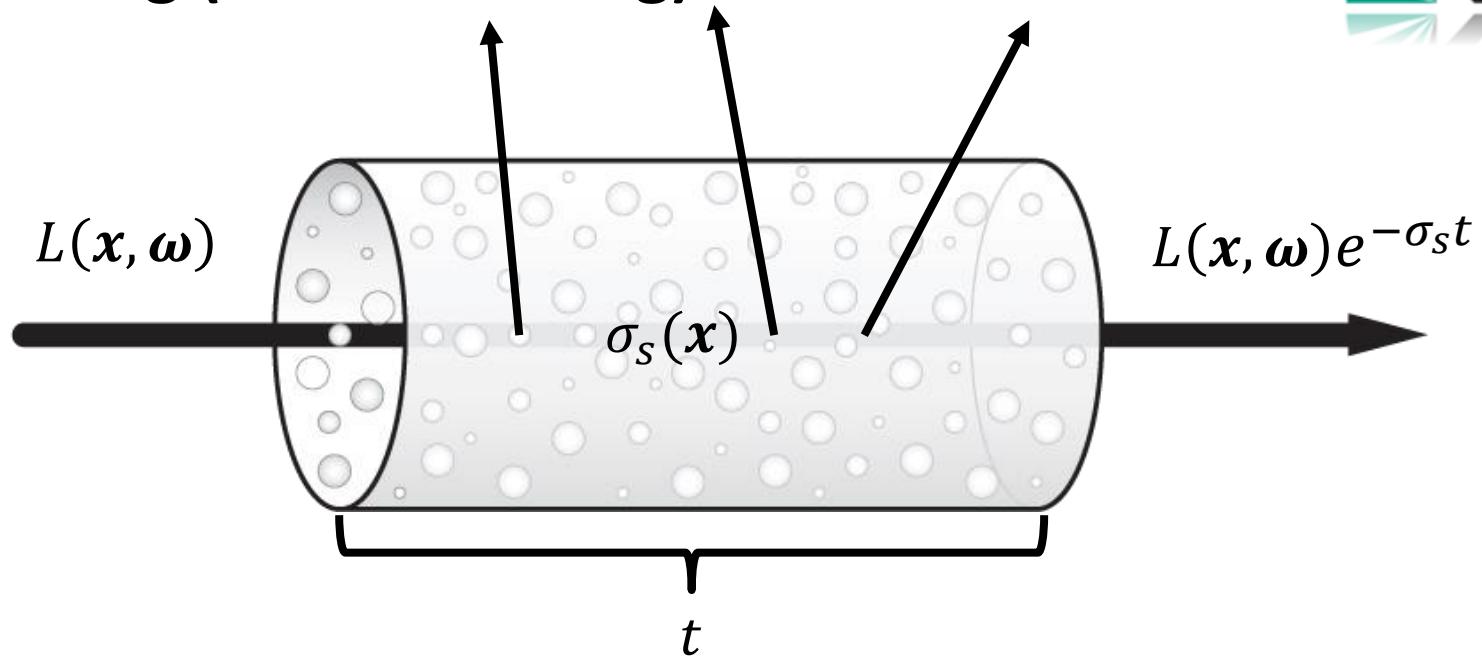
- **Absorptionskoeffizient** σ_a [m^{-1}] (hier konstant)
- Kehrwert der mittleren freien Weglänge zwischen zwei Absorptionsereignissen (Anm. $e^{-\sigma_a t}$ wg. DGL, nächste Folie)

Absorption



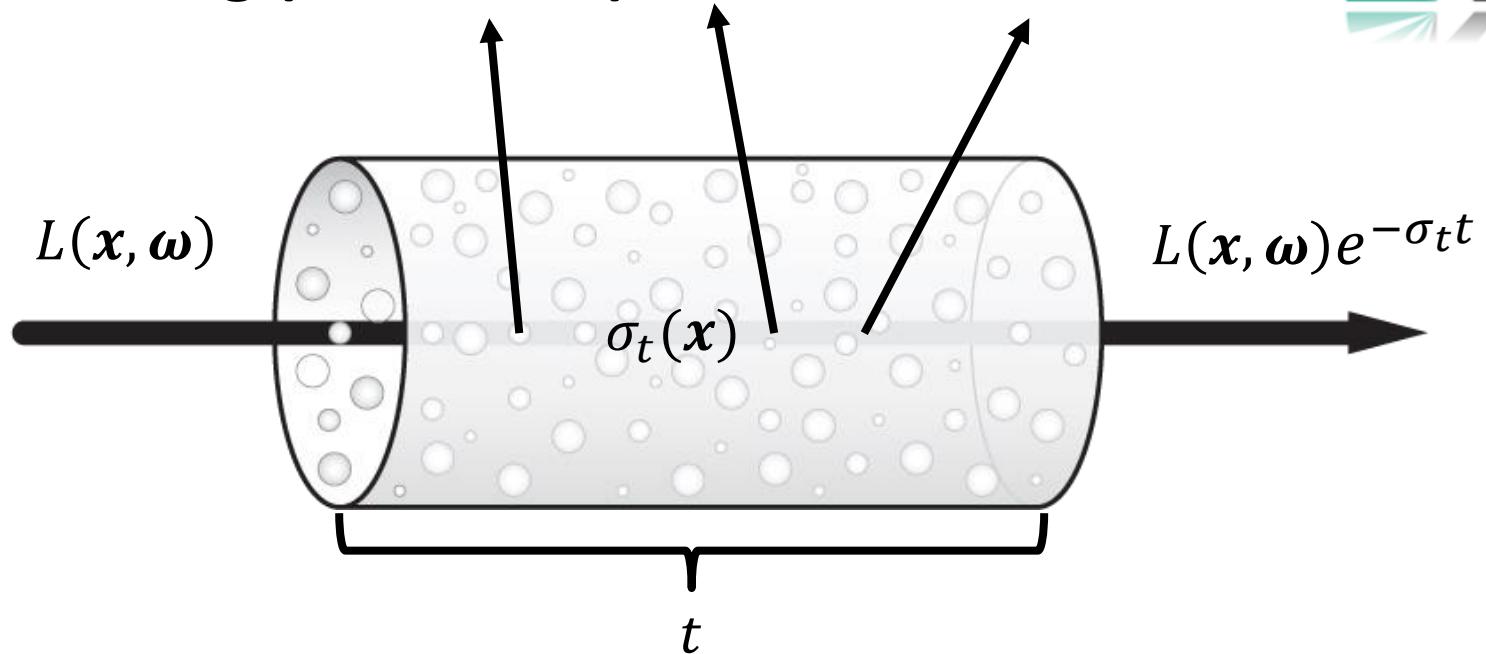
- ▶ Absorption: Änderung der Radiance an einem Punkt ist proportional zum **Absorptionskoeffizient $\sigma_a(x)$**
- ▶ entlang eines Strahls gilt:
$$\frac{d}{dt} L(t) = -\sigma_a(t)L(t) \text{ (DGL)}$$

Ausstreuung (Out-Scattering)



- ▶ der **Streukoeffizient** $\sigma_s(x)$ ist eine weitere Materialkonstante
 - ▶ „Wahrscheinlichkeit, dass ein Photon pro Längeneinheit gestreut wird“
 - ▶ für konstante Ausstreuung gilt: $L(x + t\omega, \omega) = L(x, \omega)e^{-\sigma_s t}$

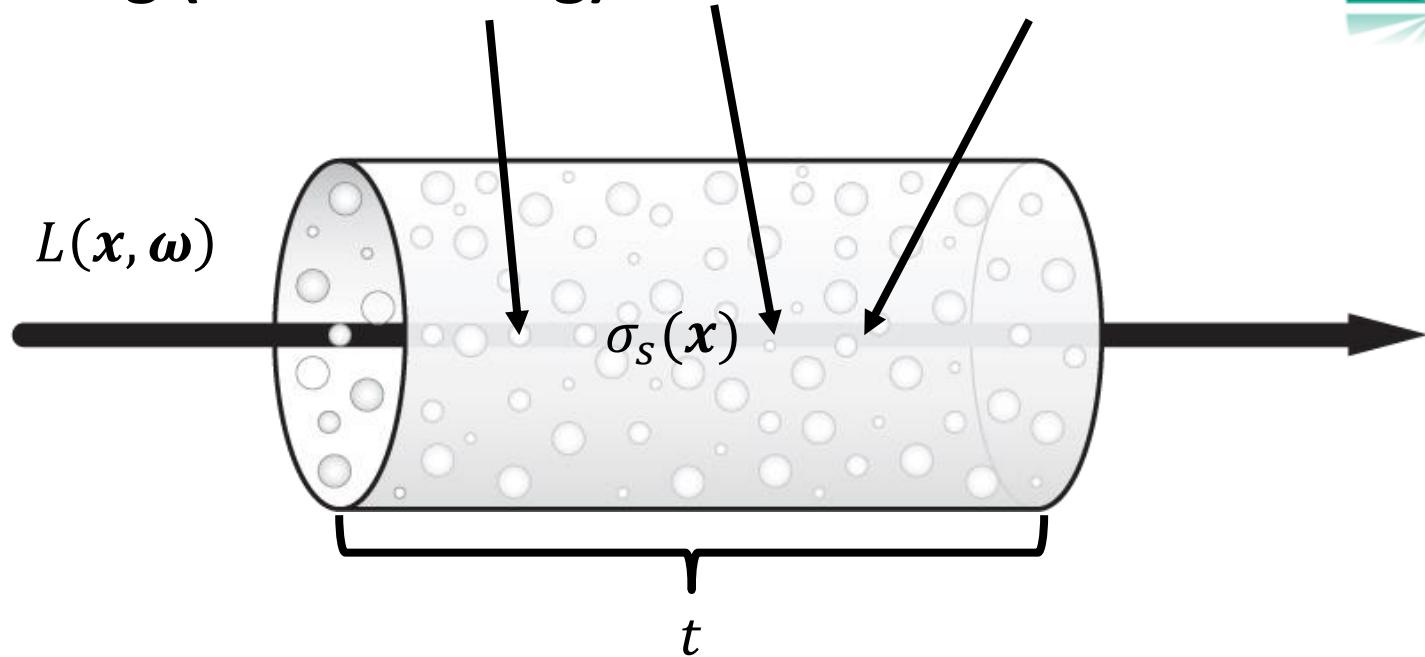
Abschwächung (Extinction)



- uns ist egal ob Radiance aufgrund von Absorption oder Ausstreuung entlang eines Strahls „verloren“ geht → zusammenfassen beider Effekte zur **Abschwächung**:

$$\sigma_t(x) = \sigma_a(x) + \sigma_s(x)$$

Einstreuung (In-scattering)



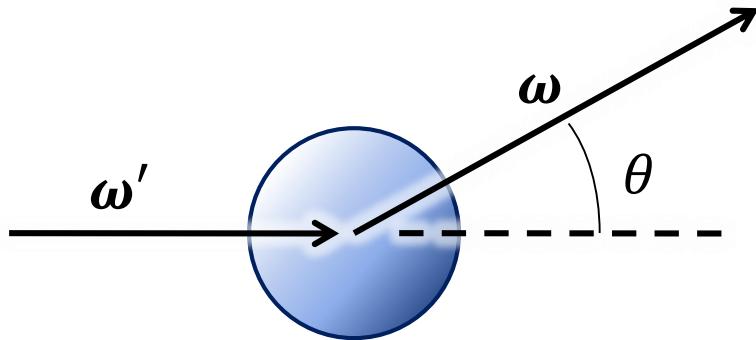
- ▶ Photonen werden auch in Richtung des betrachteten Strahls eingestreuht
- ▶ anders als Emission **hängt Einstreuung von der Radiance im Volumen ab**

$$(\boldsymbol{\omega} \cdot \nabla)L(\boldsymbol{x}, \boldsymbol{\omega}) = \sigma_s(\boldsymbol{x}) \int_{4\pi} p(\boldsymbol{x}, \boldsymbol{\omega}' \rightarrow \boldsymbol{\omega})L(\boldsymbol{x}, \boldsymbol{\omega}')d\boldsymbol{\omega}'$$

Ableitung in Richtung $\boldsymbol{\omega}$

Anisotrope Streuung, Phasenfunktion

- die Phasenfunktion $p(x, \omega' \rightarrow \omega)$ ist die Wahrscheinlichkeit, dass ein Photon an der Stelle x aus Richtung ω' in Richtung ω gestreut wird

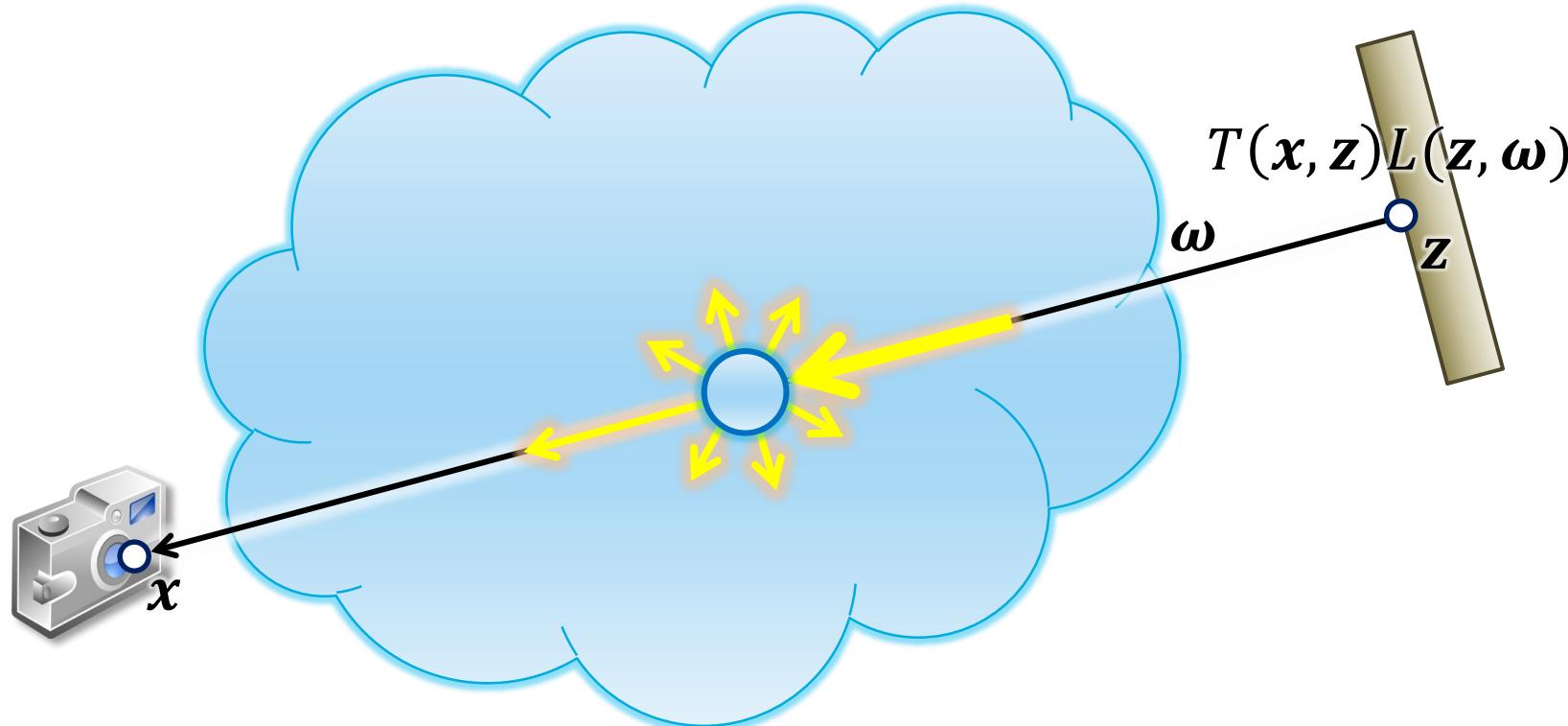


- oft hängt $p(\cdot)$ nur von Phasenwinkel $\cos \theta = \omega \cdot \omega'$ ab
(Rotationssymmetrie um ω')
- also $p(x, \cos \theta)$ oder in homogenen Medien $p(\cos \theta)$
- in isotropen Medien gilt $p(\cos \theta) = 1/4\pi$

Volumen Rendering Gleichung

- ▶ Licht wird auf dem Weg durch das Medium ausgestreut (in andere Richtungen als zur Kamera) und absorbiert
- ▶ Transmittance/Durchlässigkeit aufgrund von Ausstreuung und Absorption

$$T(x, z) = \exp\left(-\int_0^{\|x-z\|} \sigma_t(x - t\omega) dt\right)$$



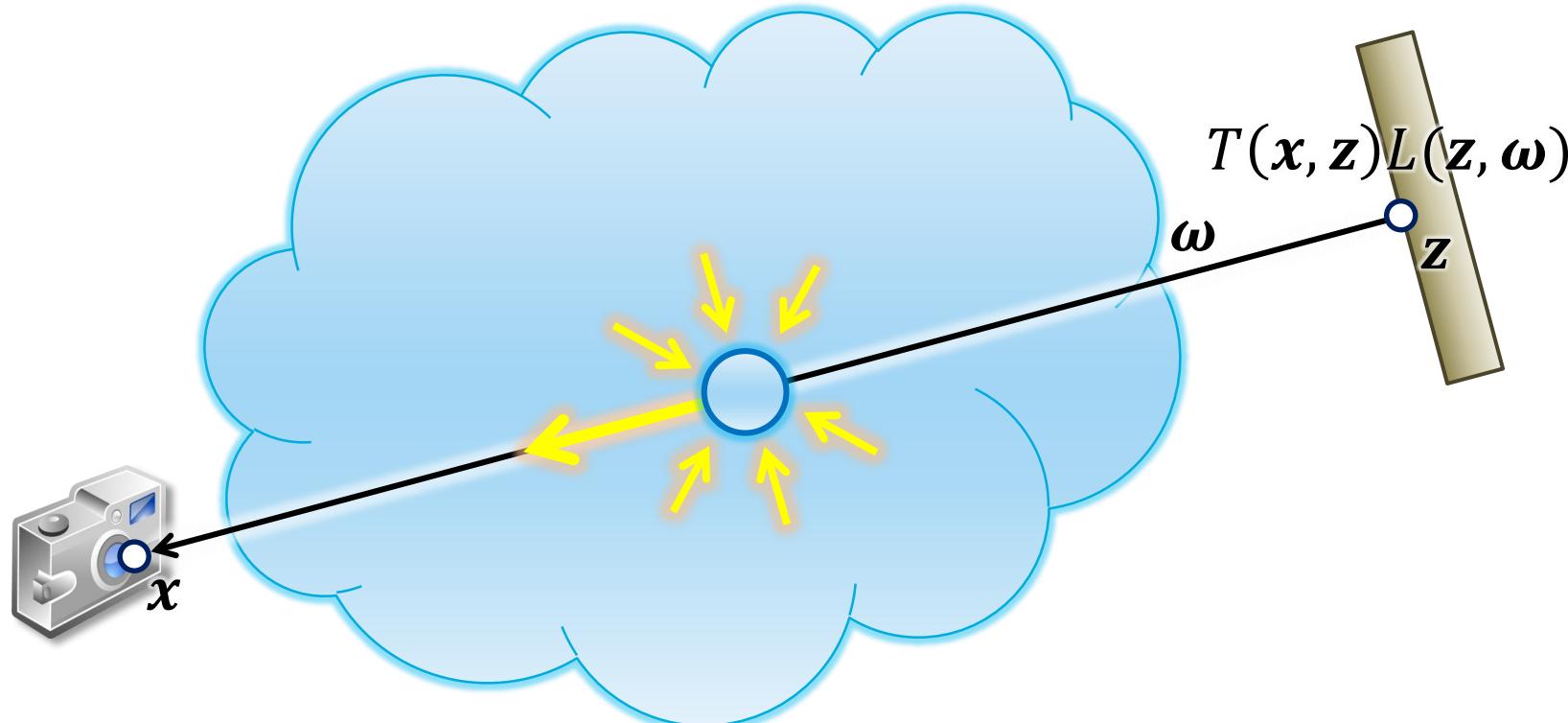
Volumen Rendering Gleichung

- Transmittance/Durchlässigkeit aufgrund von Ausstreuung und Absorption

$$T(x, z) = \exp\left(-\int_0^{\|x-z\|} \sigma_t(x - t\omega) dt\right)$$

- Emission und Einstreuung

$$S(y, \omega) = e(y, \omega) + \sigma_s(y) \int_{\Omega} p(y, \omega_i \rightarrow \omega) L(y, \omega_i) d\omega_i$$



Volumen Rendering Gleichung

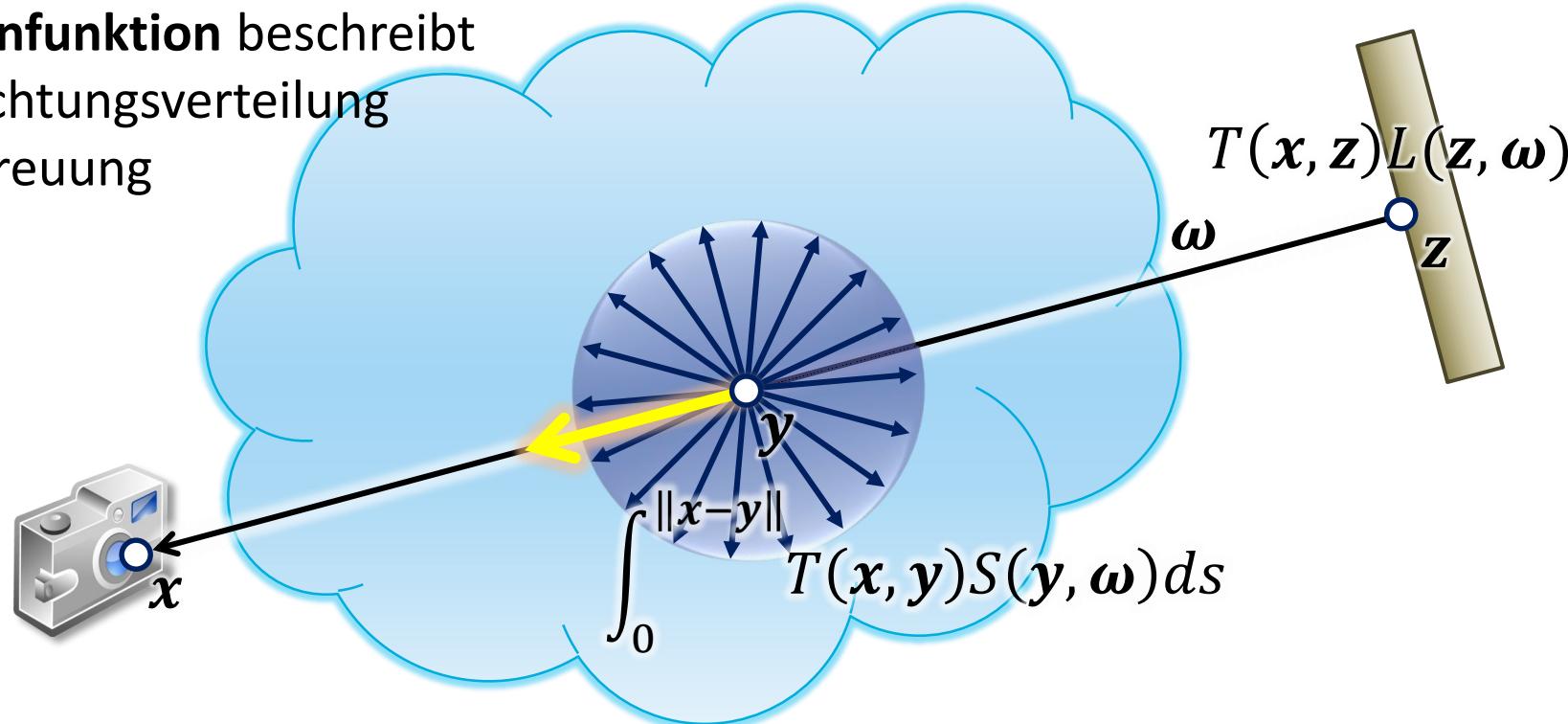
- ▶ Transmittance/Durchlässigkeit aufgrund von Ausstreuung und Absorption

$$T(x, z) = \exp\left(-\int_0^{\|x-z\|} \sigma_t(x - t\omega) dt\right)$$

- ▶ Emission und Einstreuung

$$S(y, \omega) = e(y, \omega) + \sigma_s(y) \int_{\Omega} p(y, \omega_i \rightarrow \omega) L(y, \omega_i) d\omega_i$$

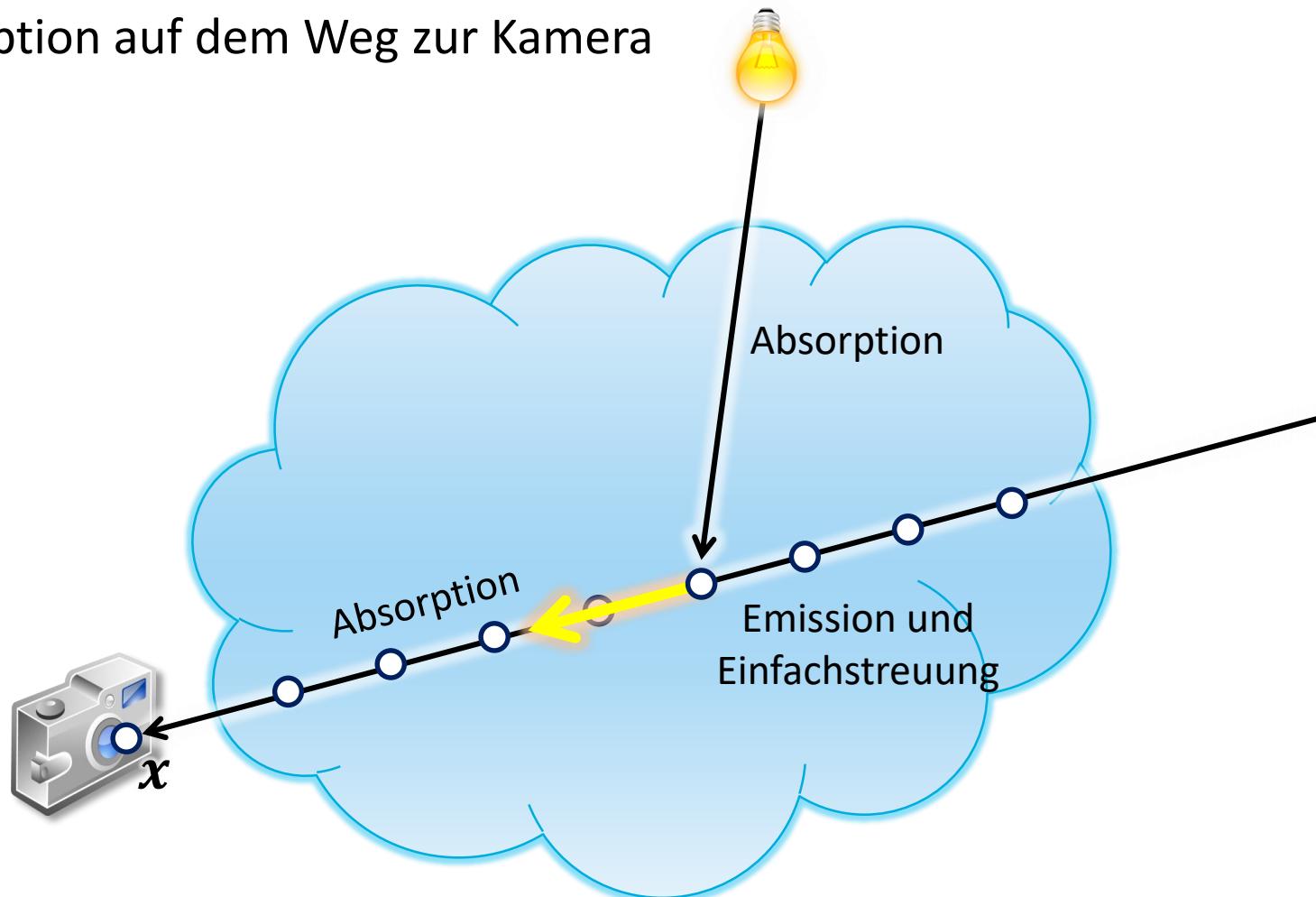
- ▶ **Phasenfunktion** beschreibt die Richtungsverteilung der Streuung



Volumen Rendering in der Visualisierung

Emissions-Absorptions-Modell

- ▶ Emission an jedem Punkt im Volumen
(optional mit Beleuchtung)
- ▶ Absorption auf dem Weg zur Kamera



Durchlässigkeitsgrad (Transmittance)

- **Transmittance** ist der Teil des Lichts, der nach einer bestimmten Strecke durch das Medium noch nicht absorbiert bzw. ausgestreut wurde

- Abschwächung/Extinction

$$\frac{d}{dt} L(t) = -\sigma_t(t)L(t)$$

- homogene DGL 1. Ordnung (Trennung der Variablen, Integration, ...):

$$\frac{dL(t)}{L(t)} = -\sigma_t(t)dt \quad \ln L(t) = -\int_0^t \sigma_t(t')dt'$$

- optische Dichte (optical thickness) $\tau(s) = \int_0^s \sigma_t(x + s'\omega)ds'$
... die Abschwächung integriert entlang eines Strahlsegments

- Transmittance

$$T(s) = e^{-\tau(s)}$$

- verbleibende Radiance

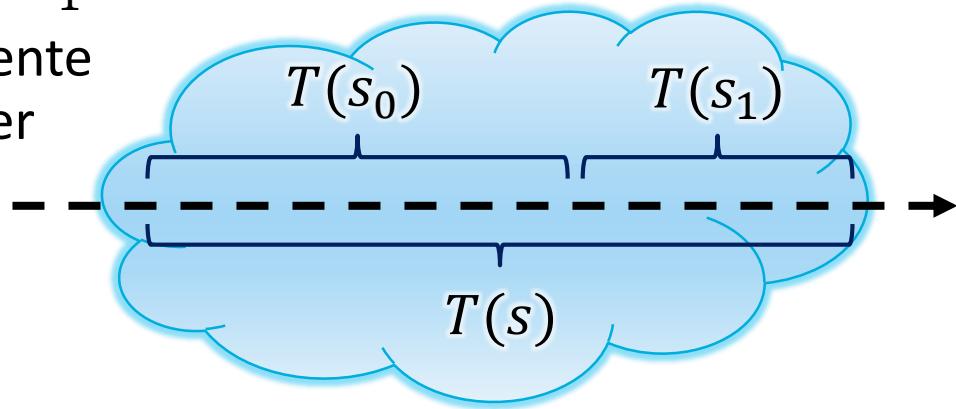
$$L(x + s\omega) = e^{-\tau(s)}L(x, \omega) = T(s)L(x, \omega)$$

Durchlässigkeitsgrad (Transmittance)

- ▶ Transmittance verhält sich multiplikativ für Teilstücke eines Strahls, d.h.

$$T(s) = T(s_0) \cdot T(s_1) \text{ mit } s = s_0 + s_1$$

- ▶ dadurch können einzelne Segmente entlang des Strahls nacheinander betrachtet werden



- ▶ Spezialfall: homogenes, isotropes Medium mit $\sigma_t(t) = \sigma_t$

- ▶ Beersches Gesetz (dieser Spezialfall) besagt: $T(s) = e^{-\sigma_t s}$

- ▶ ergibt sich aus der optischen Dichte, die zu $s \cdot \sigma_t$ wird

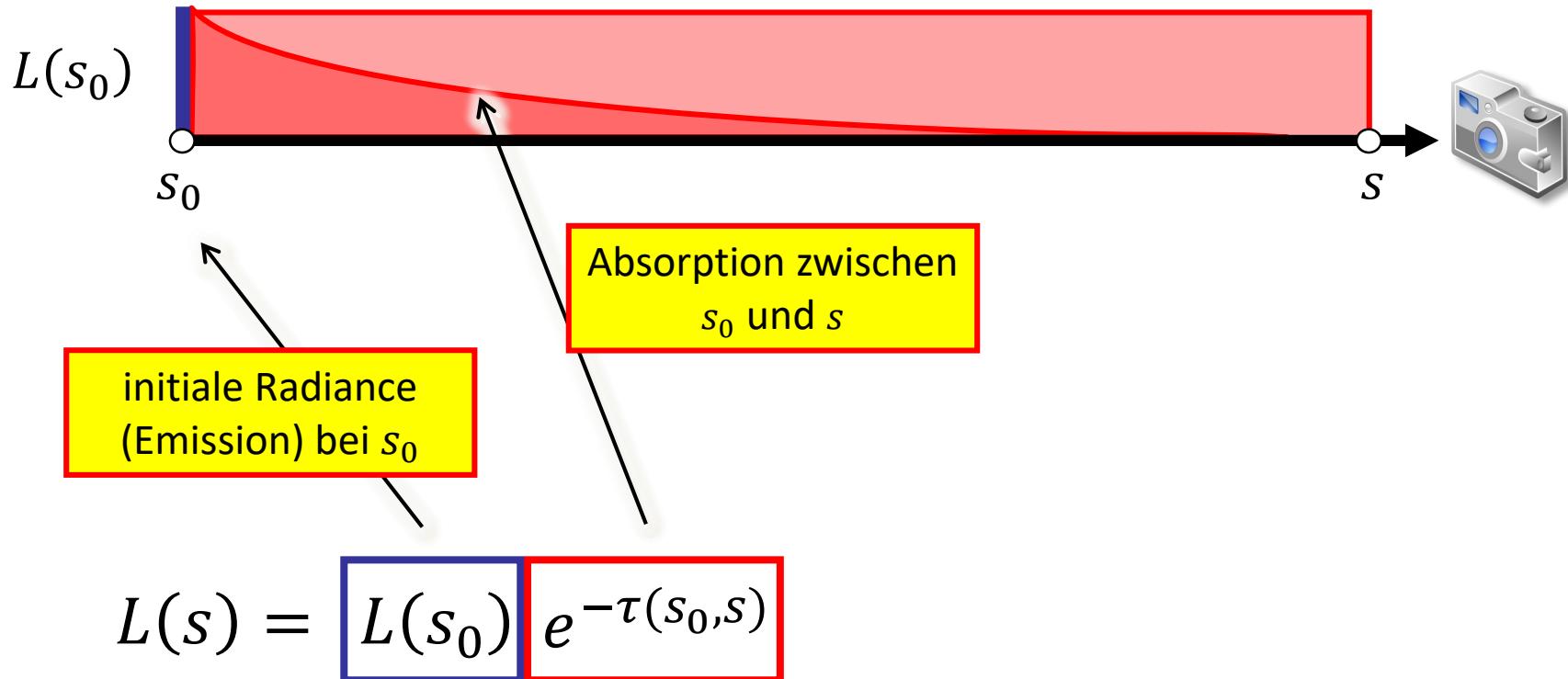
- ▶ Beispiel: $s = s_0 + s_1$, σ_t const.

$$\Rightarrow T(s) = e^{-\tau(s_0+s_1)} = e^{-\tau s_0} e^{-\tau s_1} = T(s_0)T(s_1)$$

Volumen-Rendering Gleichung

Emissions-Absorptions-Modell

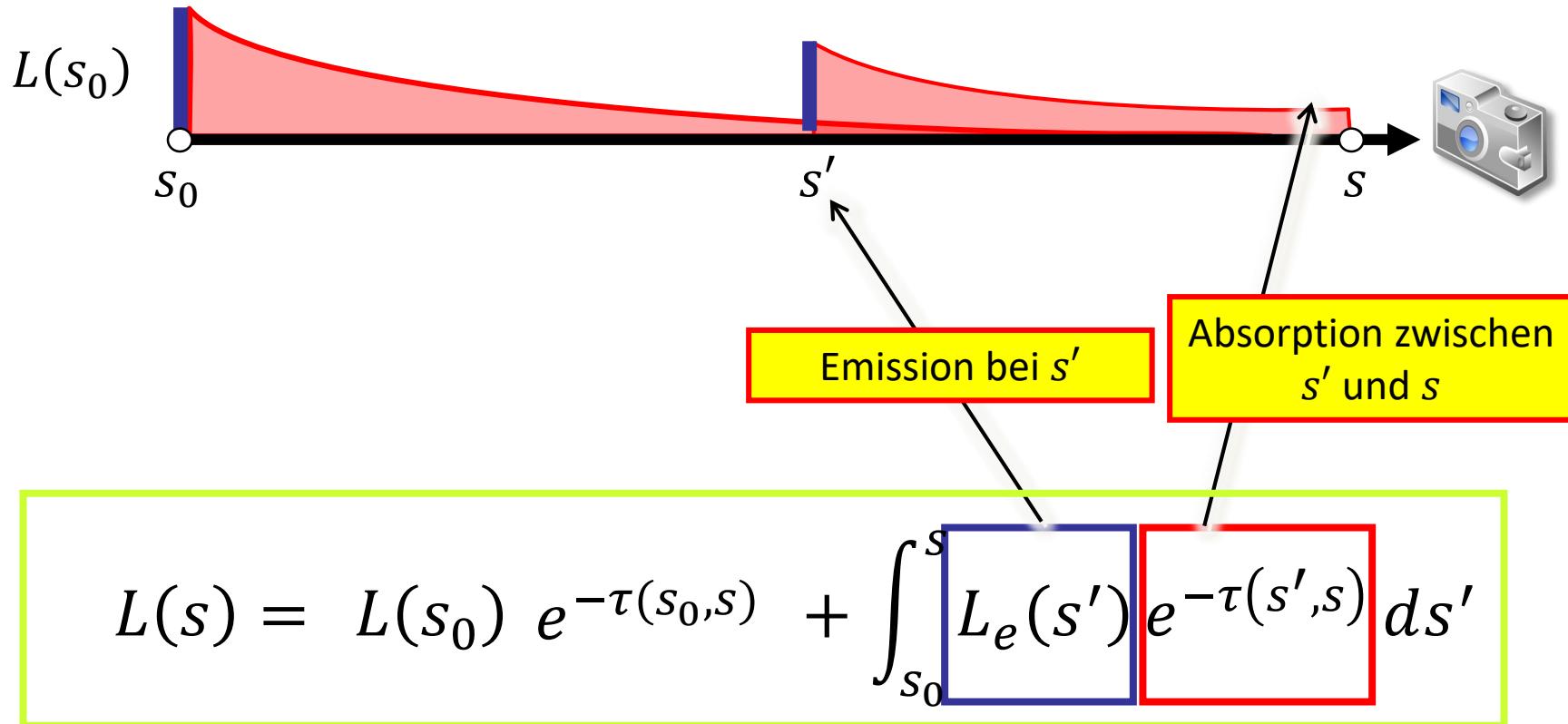
- ohne Absorption würde die Anfangsintensität ohne Verluste am Punkt s ankommen



Volumen-Rendering Gleichung

Emissions-Absorptions-Modell

- an einem Punkt s' entlang des Strahls wird zusätzlich Licht emittiert



Volumen-Rendering Gleichung

- ▶ Berücksichtigung aller Effekte (Absorption, Emission, Ein-/Ausstreuung) ist meist zu aufwendig für die Darstellung
 - ▶ ... und oft auch gar nicht gewollt, z.B. Mehrfachstreuung verhindert klare Darstellung in semi-transparenten Bereichen
- ▶ das klassische **Emissions-Absorptions-Modell** (oft auch als Density-Emitter Model nach [Sabella 1988] benannt) berücksichtigt nur
 - ▶ Emission (aus TF), keine Einstreuung: $L_e(\mathbf{x}, \omega)$ oder $L_e(\mathbf{x})$
 - ▶ Absorption: $\sigma_t(\mathbf{x}) = \sigma_a(\mathbf{x})$

Volumen-Rendering Gleichung

- ▶ ExposureRender: Lösung der vollen Volumen-Rendering Gleichung mit Monte Carlo Methoden und CUDA
 - ▶ <http://graphics.tudelft.nl/Publications/Kroes2011>
 - ▶ <http://code.google.com/p/exposure-render/>

