

# Visualisierung Medizinischer Daten 2

Zusammenfassung WS15/16

## Vorwort

Hallo, ich schreibe solche Zusammenfassungen eigentlich nur für mich selbst, da ich schon beim Akt des Schreibens einiges lerne. Da es eigentlich NUR für mich (und vll den ein oder anderen Kollegen) gedacht ist, ist das ganze enorm schlecht geschrieben und voller Tippfehler etc. Teils sind mir während dem Schreiben auch einige Passagen nicht ganz klar, dass manifestiert sich dann natürlich auch wieder im Text in Form schwammiger Formulierungen.

Wenn du bereit bist für den Dreck, und es dennoch als angenehm empfindest ein zentrales Dokument zu den Folien zu haben ... viel Spaß hiermit. Lg AllesBeimAlten (InfForum)

## Kapitel 1: Medical Imaging Data: Acquisition & Integration

### Acquisition Techniques

Generelle zu Berücksichtigende Modalitäten sind die **Aussetzung an Radioaktivität, Kosten & Dauer** der Erstellung und **Beschränkungen in der Anwendung**.

#### Charakteristische Dimensionen

1. Räumliche Auflösung (Pixelgröße)
2. Zeitliche Auflösung (Scanzeit)
3. Bild Matrix (Anzahl an Pixeln)
4. Messungen (Projektion, Volumen)

#### Röntgen

Alte Technik, Dämpfung von Röntgen beim Durchschreiten verschiedener Gewebsformen wird gemessen. Dämpfung abhängig von Dichte der Röntgenstrahlen und Gewebsdichte.

2 Arten von Strahlen:

- Hard Beam (Knochen, > 100kV)
- Soft Beam (Weiches Gewebe, schädlicher)

#### Angiographie

Zur Darstellung von Gefäßen, Röntgen (oder CT) basiert. Radioaktiver Agent wird injiziert, Katheder ins arterielle System eingeführt.

**Digital Subtraktion Angiographie (DSA)** ist ein verwandtes Verfahren. Funktioniert per X-Ray und stellt **Blutgefäße obstruktionsfrei** dar. Zuerst wird ein **klassisches Röntgen** aufgenommen, dieses dient als Basis (Mask Image), **alle darauffolgenden Bilder** werden unter **Einfluss eines radioaktiven Agenten** durchgeführt und dem Mask Image **abgezogen**. Es bleiben nur die Blutgefäße.

DSA ist hoch räumlich und zeitlich aufgelöst, lässt aber keine Schlüsse über etwaigen Plaque oder das umgebenden Gewebe der Blutgefäße zu.

#### Einsatz:

1. Stenosen
2. Thrombosen

### 3. Aneurysmen, etc.

#### Computertomographie

Äquidistante Schnitte, Seit den 70ern, basiert auch auf Röntgenstrahlung. Röntgenquelle rotiert um Patienten erstellt 1D-Dämpfungsbilder, Zurückrechnen auf Bild per Radontransform. Hounsfield Unit:

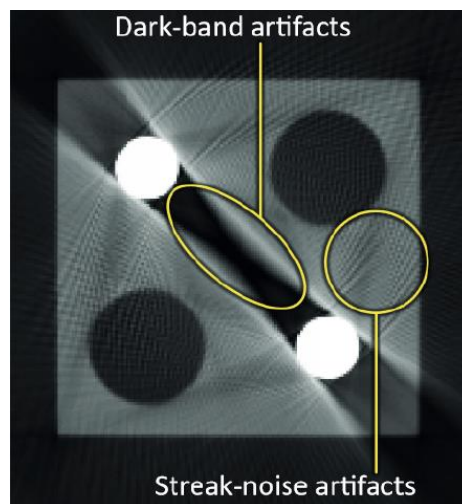
$$HU(x, y, z) = 1000 \cdot \frac{\mu_{x,y,z} - \mu_{Wasser}}{\mu_{Wasser}}$$

HU von Knochen 1000+, Wasser hat 0, Luft -1000.

**CT Pros:** Geringe Slice Distanz (hohe räumliche Auflösung), Kann große Blutgefäße und Knochenbrüche erfassen

**CT Cons:** Geringer Kontrast bei weichem Gewebe / unterschiedliche Gewebetypen teils schwer unterscheidbar, Röntgenstrahlung (Invasiv)

**Artefakte:** Metall erzeugt Artefakte und folgende beide:



**Hochauflösend:** Bei einem hochaufgelösten CT beträgt die Auflösung 1024x1024 (gegen 512x512), selbe Scannprozedur – anderer Rekonstruktionsalgorithmus.

**DECT:** Dual-Energy CT – hat zwei Quellen und Detektoren mit 90° Offset. Die Energie von beiden Quellen ist unterschiedlich. Gewebe zeigen unterschiedliches Dämpfverhalten bei unterschiedlicher Energie → lässt z.B. automatische Separation von Knochen/Gewebe zu. Wird angewandt zur Segmentation von Knochen, kann auch Artefakte eliminieren.

**Niedrige Energie:** Hoher Gewebesartkontrast, mehr Noise. **Hohe Energie:** Weniger Gewebesartkontrast, weniger Noise.

**CT Angiographie (CTA):** Kontrastmittel wird in venöses System injiziert, nicht arteriell wie bei DSA. Blutgefäße werden trüber oder sichtbarer.

**Kontrast Agenten Injektionsprotokolle:** Blut fließt mit unterschiedlichen Geschwindigkeiten durch verschiedene Organe, führt zu mehrfacher „Trübung“ wegen Rezirkulation. (Herz)

- **Uniphasisch:** Injektion von Kontrastmittel könnte zu nicht-uniformen Vessel-Enhancement führen.

- **Biphasisch:** Zuerst eine konstante Menge an Kontrastmittel, dann sukzessive Senkung der Menge. (Führt zu nahezu Uniformer Verteilung)
- **Multiphasisch**

## MRI

Auch seit den 1970, mit Magnetfeld insofern non-invasiv. Hohe Auflösung, auch bei weichem Gewebe. Kann „frei wählbare“ Querschnitte erstellen. Eingesetzt zur Tumorerkennung, fMRI (Gehirnfunktionen), Funktionale Diagnostik, ...

**Prinzip:** Starkes Magnetfeld → Anregung von Wasserstoff (Spins) durch RF-Pulse → Relaxation der Atome sendet Energie aus (Gewebsabhängig) → Energie wird gemessen

- **T1 (Logitudinal):** Zurückflippen des Spins
- **T2 (Transversal):** Zurückkehren in Phase

**T1 und T2** erzeugen unterschiedliche Bilder.

MRT	CT
Teuer	Billig
Sehr flexibel	Statischer
Guter Weichgewebe Kontrast	Generell schlechterer Kontrast
fMRI	Blutungen gut erkennbar
Eher langsames Scannen	Schnelles Scannen
Komplexer, erfordert Erfahrung und Tests, viele Parameter	Einfach

## PET Scan

Radioaktive Substanz als Marker, Zerfall wird gemessen. (511keV, gegengesetzte Richtung) Gut zur Darstellung physiologischer Prozesse. Schlechte räumliche Auflösung: 1-3mm. 64x64/128x128.

**Einsatz:** Beobachtung von Gehirntumoren, Epilepsie, Kardiologie.

## Ultraschall

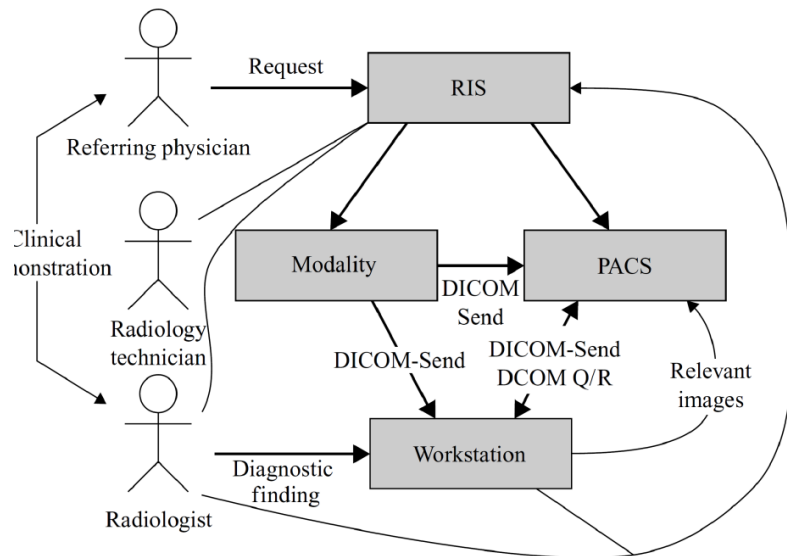
„Transducer“ schickt Schall durch Gewebe. Ziemlich billig und non-invasiv, dafür noisy, geringer Kontrast, unscharfe Grenzen, komplex zu interpretieren, kein standardisierter Export. (DICOM)

Einsatz: Fetalbeobachtung (+Rendering), Tumor im Abdomen, Intra-operative Navigation.

## Integration

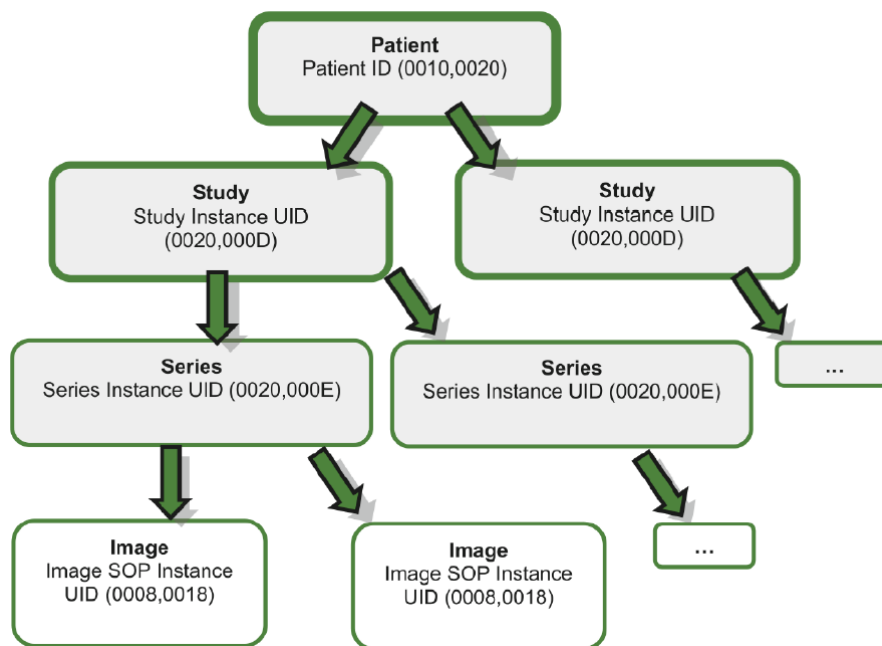
Das Hospital Information System (HIS) verarbeitet alle Informationen und Prozesse in einem Spital.

Subsysteme: Picture Archiving and Communication System (PACS), Radiology Information System. (RIS)



## DICOM

Digital Imaging and Communications in Medizin, Protokoll zum Austausch / zur Verarbeitung medizinischer Bilddaten. Syntax und Semantic definiert. DICOM-kompatible Geräte unterliegen dem Protokoll.



## Struktur von Dicom Files

- Dicom Meta Informationen
- Dicom DataSet / Nachricht
  - Eine Sequenz von Daten-Elementen
  - Geordnet nach Gruppe und Data-Element-Nummer
- Dicom Data Element
  - Tag: Gruppen Nummer (8 Bit), Data Element Nummer (32 Bit)
  - Datenlänge (32 Bit)
  - Daten

**Gruppen-Nummern: Gerade Nummern:** Infos zu Patient, wie die Daten erstellt wurden, Präsentationsdaten, Studiennummer, Studienzeit, Modalität, Patient Info, ... etc. **Ungerade Nummern:** Vendorabhängig.

■ Identifying information:

(0008,0020)	DA	Study Date	# yyyy.mm.dd
(0008,0030)	TM	Study Time	# hh.mm.ss.frac
(0008,0060)	CS	Modality	

■ Patient Information:

(0010,0010)	PN	Patient Name	
(0010,0030)	DA	Patient Birth Date	# yyyy.mm.dd
(0010,0040)	CS	Patient Sex	# M, F, O for other

Tag      Value      Attribute Name  
Rep.

**Windowing Function:** 12 bit Daten werden auf 8 bit gemappt, Kontrastregulierung

**Protected Health Information:** Ist ein Term im „US Law“, beschreibt Informationen über Gesundheitsstatus und weiteres dass zu einer Person gelinkt werden kann. Wenn Wissenschaftler einen derartigen Datensatz veröffentlichen versuchen sie PHI zu entfernen.

- **De-identification:** PHI wird entfernt, externe Link zu einer vertrauenswürdigen Ressource wird erhalten, die in der Lage ist öffentliche Daten und PHI wieder zusammen zu führen.
- **Anonymization:** PHI wird gänzlich entfernt. Keine Rekonstruktion möglich.

## Kapitel 2: Segmentierung

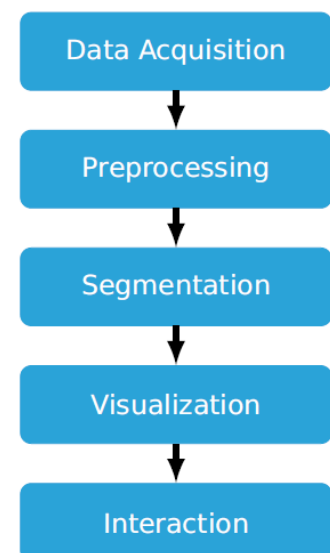
### Preprocessing

Hierbei werden die Daten zu weiteren Verarbeitung „verbessert“, durch Operationen wie Noise Reduction, Edge und Contrast Enhancement etc.

**X-Modalität von Daten:** Bei Uni-Modalen Daten liegt ein distinktives Maxima vor, bei Bi-Modalen 2, bei Tri-Modalen 3, ...

#### Arten von Preprocessing

1. **Punktorientiert:** Voxel werden isoliert betrachtet. Thresholding, Subtraktion etc.
2. **Lokal:** Nachbarschaftsinfo wird benutzt. Statische/Dynamische Filter. Problem: Kantenbehandlung.
3. **Global:** Alle Pixel werden betrachtet. Histogramme, Fourieranalysen.



CT & X-Rays: Haben oft normal verteiltes Noise

#### Filtertypen

Filter	Funktion	Nutzen	Eigenschaften
--------	----------	--------	---------------

Mittelwertfilter	Mittelt Werte über Nachbarschaft	Noisereduction	Linear, „Plättet“/Glättet Kanten, führt neue Werte ein.
Medianfilter	Mittlerer Wert der Nachbarschaft wird bestimmt (Sortierung + Auswahl)	Noisereduction	Non-Linear, erhält Kanten jedoch keine sehr dünnen Linien und „Outliers“. Ändert den Durchschnitts Grauwert des Bildes, führt keine neuen Werte ein.
Gauss-Filter	Wie Mittelwertfilter nur gauss-gewichtet.	Noisereduction	Non-Linear, Gut für Noise in kleinen Bereichen, glättet Kanten, CT & X-Ray haben normalverteiltes Noise
Sigma-Filter	Sigma bestimmt Weite 1. Berechne Standardabweichung "stddev" innerhalb Kernel 2. Für jeden pixel/voxel p im Kernel dessen Intensität zwischen $-\sigma \cdot \text{stddev}$ und $+\sigma \cdot \text{stddev}$ liegt berechne Average und setze für Pixel.	Noisereduction	Non-Linear, Filtert nur Pixel/Voxel die leicht vom Durchschnittswert abweichen
Bilateraler-Filter	Gewichteter Average Abhängig von 1. Distanz der Pixel vom Zentrum 2. Intensitätsunterschiede $\sigma_r$ : (Range, Intensitätsunterschied) Umso größer Parameter umso enger (Range unterschied wird irrelevanter) $\sigma_d$ : (Spatial, Entfernung) Umso größer, umso größere Flächen werden geslätet	Noisereduction	Non-Linear, Kantenerhaltend, Gewichtung Gaussbasiert (Modifikation von Gausskernel nach Distanz / Intensität), Sehr rechenintensiv. Bei DECT: Low Energy Level für die Intensitätswerte, „Edgeweight“ per High Energy-Level
Contrast Verbesserung	Berechne Laplace-Kantenbild, Differenziere zu von Originalbild	Kontrast erhöhen bei Objekt-Kanten	
Schärfen	Kernel mit negativen Werten in Nachbarschaft gegen starkpositive Werte in Zentrum, Summe von ganzen Kernel ist 1	(Kanten)schärfung	
Unsharp Masking	Generation von stark geglätteten Bild, Differenziere zu Originalbild	Entfernung unscharfer Regionen	

**Generell:** Erhöhung des Kontrastes = Erhöhung des Noise, Reduzierung des Noise = Reduzierung des Kontrastes

## Segmentierung

Identifikation von Pixel/Voxeln **nach bestimmten Homogenitätskriterien**. Wird benötigt um:

1. Bestimmte Objekte zu betonen
2. Klassifikation (Knochen, Gefäße, ...)
3. Weitere Verarbeitung wie Skeletonization

#### 4. Quantitative Analysen (Volumen, Formen, ...)

**Abhängig von:** Modalität (CT, MRI, ...), Scan-Protokoll, Objekt (Dünnes Gefäß oder großer Organ?)

Thresholding:

Pixel/Voxel einfach nach Intensität klassifiziert.

**Pro:** Schnell simpel, akzeptable Ergebnisse schnell erreicht (gut für den Anfang)

**Cons:** Räumliche Beziehungen werden ignoriert, im Endeffekt zu schlecht für wirklich akzeptable Resultate

Um getrennte Regionen zu verbinden (und/oder die größte verbundene Komponente zu finden) wird meist **Connected Component Analysis (CCA)** eingesetzt.

Varianten: **Single threshold**, **Two thresholds** (Objekt zwischen beiden Thresholds, alles andere ist Foreground/Background) und **lokale Techniken** (z.B. basierend auf Histogramm-Analyse)

Region Growing

**Seedpunkt** als Start, Region wächst nach **Inklusionskriterium** (Threshold-basiert, Similaritäts-basiert). Problem ist das „leaken“ durch Löchern in der Kontur

**Verbesserungen:**

1. Vom User festgelegte Grow-Richtung
2. Adaptive Thresholds
3. Berücksichtigung von Gradient(magnitude)
4. Einbezug von Vorwissen

Morphologische Operationen

Erosion (Reduziert) und Dilation (Vergrößert). Benötigen ein Strukturelement (Kreis, Sphäre etc.), welches bei Erosion versucht in das Objekt zu „fitten“ und bei Dilation „aus dem Objekt“ gelegt.

**Öffnung:** Erosion → Dilation, **Schließung:** Dilation → Erosion

Skeletonization

Erlaubt die Beschreibung der Form eines Objektes. Das Skelett verläuft hierbei exakt in der Mitte bzw. dem Zentrum des Objekts und ist im diskreten Fall exakt 1-Pixel bzw. Voxel breit. **Einsatz** für Messungen, Pfadplanung, Branch-Analyse und diverse Visualisierungen (Bronchien, Blutgefäße)

Algorithmen:

1. **Centers of Maximum inscribed Circles/Spheres**
2. **Continued Erosion, Thinning**
3. **Grassfire transform** (Skeleton remains after „burning“, Boundary on Fire)

Probleme:

1. Irrelevante Sidebranches
  - a. Gegenmittel: Pruning, Irrelevante Branches werden nach Kriterien weggefilter: Länge, Verhältnis Länge Sidebranch zu Mainbranch, Hierarchische Position (Leafbranches werden entfernt)
2. Zwei-Pixel-weite/Zu dicke Linien

## Segmentationsprobleme und Qualitätskriterien

### Probleme:

1. Untersegmentation
2. Obersegmentation
3. Intensitätsunterschiede im Organ wegen Krankheit
4. Nonuniforme Kontrastverbesserungen
5. Scan-Artefakte

Müssen meist **interaktiv vom Benutzer korrigiert** werden. z.B. per Thershold-Korrekturen, Limitation der automatischen Segmentierung zu bestimmen Regions-of-Interest und „Stepwise Refinement“.

### Qualitäts-Kriterien:

1. **Geschwindigkeit** (Anwendung Abhängig: Im Notfall schnell einsetzbar?)
2. **Robustheit gegen Noise aller Art:** Metalle, Bewegungen des Patienten, Pathologische Veränderungen
3. **Schnelles Feedback**
4. **Reproduzierbarkeit**

**Präzision** ist gegeben durch Vergleich der aktuellen Lösung an einer Referenzlösung (Modell von Hersteller oder erstellt von Experten). Weitere Kriterien: Jaccard-Coefficient (Similaritätsmaß), Falsch-Positive gegen Falsch-Negative

### Quantitative Analyse

Analyse wird zusätzlich zu „Visual Reviewing“ durchgeführt. Erlaubt objektivere Diagnosen und Verlaufsanalysen und hilft dadurch bei der Entscheidungsfindung. (Man denke an die Gelenkskrankheit aus MedBildVer)

### Mögliche Messungen:

1. Tumoranzahl und Größe
2. Winkelzwischen anatomischen Strukturen
3. Distanz zwischen Läsionen und risikobehafteten Organen
4. Intensitätsverteilungen des Objekts (Mittelwert, Stddev)

### Feststellung der Maße eines Objekts

Naive Idee hierzu wäre einfach eine achsen-alignte Bounding Box um das Objekt (Tumor, wasauchimmer) zu legen und einfach die min/max Werte entlang der Achsen zu bestimmen. Probleme gibt's jedoch hierbei wenn das Objekt rotiert ist, somit ist eine achsen-alignte Bounding Box nicht Mittel der Wahl.

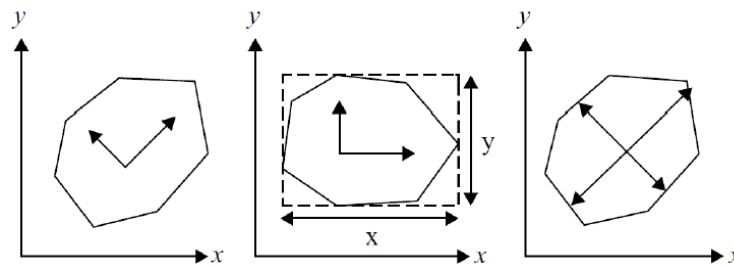
Bessere Idee: Hauptachsentransformation, Bounding Box um Hauptsachen des Objekts legen.

### Algorithmus:

1. **Berechnung des Baryzentrumvektors  $m$**
2. Subtrahiere  $m$  vom Objekt (**Zentrierung**)
3. **Berechne Kovarianzmatrix  $A$**
4. **Berechnung der Jakobi-Matrix  $C$**  (Normalisierte Eigenvektoren von  $A$ )
5. **„Virtuelle“ Objektrotation** nach  $C$
6. **Berechnung der achsen-aligten Bounding Box** des rotierten Objekts um Länge der Hauptachsen festzustellen



## 7. Rücktransformation von Objekt zu $C^{-1T}$ mit Transposition



### Visualisierung von Segmentierung

Standard Approach wäre einfach per Binary-Maske (Background, Foreground). Alternative: **Marching Cubes**, hierbei können jedoch Treppenartefakte entstehen bei großer Slice-Distanz. Lösung: Kubische Interpolation.

## Kapitel 3: Indirect Volume Visualization

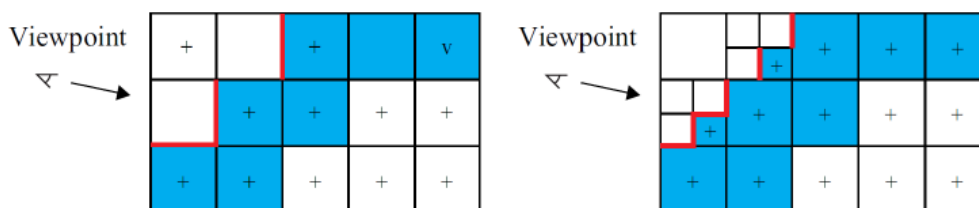
Warum indirekte Visualisierung? Oberflächenmodelle sind z.B. interessant zur Veranschaulichung (medizinische Ausbildung), Interventionsplanung und Diagnose. Außerdem gibt es Stoffe die schwer direkt visualisierbar sind. (Gas, Flammen, Flüssigkeiten)

Der **Input** bei indirekte Volumensvisualisierung sind bereit segmentierte Daten relevanter Strukturen. (Tumore, Organe z.B. in Binary-Repräsentation)

## Techniken

### Cuberrile

Sehr simple Herangehensweise wobei aus den Voxeln (minecraft-ähnliche) Blöcke erzeugt werden. Die Oberflächen sind sehr grob. (eben Blockig) Verbesserungen durch Glättungsalgorithmen und „Subdivide“.



### Marching Cubes

Eine Kontur muss in 2D bestimmt werden, der Kontur wird gefolgt und das Objekt in 3D verbunden. Probleme Damit: Schwerfälligkeit und zu viele „Fälle“ die Berücksichtigt werden. **Idee:** Würfel wandert durch Volumen. Zellen werden lokal behandelt und Schnittpunkte der Zellen mit den Oberflächen bestimmt, woraus die lokale Oberflächenform resultiert.

### Problemefälle: (Surfe Extraction Generell)

1. **Korrespondenz zwischen Konturen aufeinanderfolgender Slices**
2. **Triangulation:** Wie wird das Triangel-Mesh zwischen zwei Konturen erstellt?
3. **Branching points:** Wie verbindet man eine unterschiedliche Zahl an Konturen in zwei aufeinanderfolgenden Slices

4. Was soll getan werden, wenn zwei Konturen **aufeinanderfolgender Slices zum selben Objekt gehören, sich aber nicht überschneiden?**

Anforderung zum Einsatz von Marching Cubes: Überlappende Strukturen innerhalb aufeinanderfolgender Slices.

**Problematisch bei Marching Cubes** ist eine weite inter-Slice-Distanz bei dünnen Objekten, die diagonal zur Oberfläche positioniert sind. **Lösungen:** Alle Teuer: Generalized Elliptical Cylinders, Interpolation von Zwischenslices.

Um die **Performanz zu verbessern** werden alle möglichen **Fälle** von Konturverläufen durch die Zellen bei Marching Cubes **als Binärrepräsentation dargestellt**. Die 256 Fälle lassen sich hierbei durch **Symmetrien auf 15 Fälle reduzieren**. Bei Mehrdeutigkeiten kann der Benutzer manuell entscheiden. (vll eher nur im 2D Fall)

#### Ablauf:

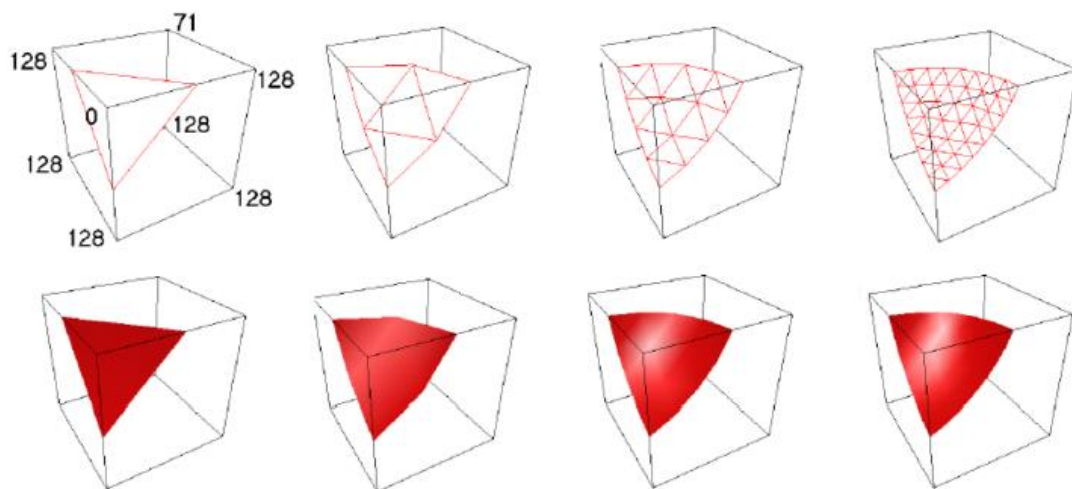
1. Oberflächenfall für jede Zelle bestimmen
2. Für jeden Edge-Cut linear interpolieren
3. Verbinde die Schnittpunkte zu Dreiecken
4. Berechne Normalvektoren

Qualitäts-Probleme bei den Surfaces können per linearer Interpolation und Gouraud Shading aufgehoben werden.

**Pro:** Simpel und Schnell **Cons:** Teils sichtbare Artefakte, Viel Zeit wird bei Zellen vergeudet die nicht auf der Oberfläche liegen.

#### Precise Marching Cubes

Basiert auf Marching Cubes, die Oberfläche wird jedoch adaptiv verbessert.

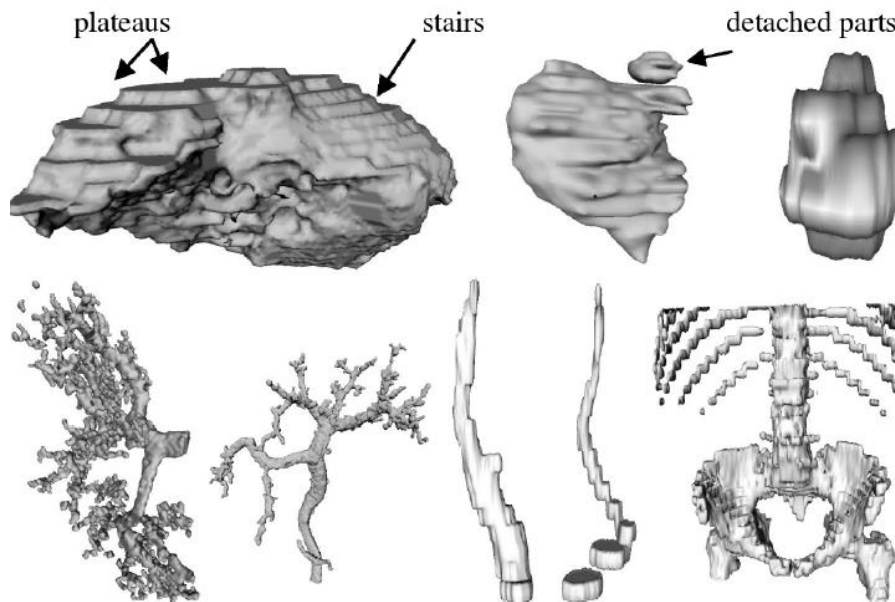


Es ist 100x aufwändiger als normales Marching Cubes und die Oberfläche besteht aus 2-3x so vielen Dreiecken.

Die Dreiecke werden **rekursiv geteilt solange** der **Fehler über einen gewissen Threshold** liegt. Der **Fehler** wird bestimmt indem man den **Abstand der Dreiecke zur Oberfläche** an bestimmten Samplepunkten misst. (Zentrum, Ecken) Steigt der Fehler über ein bestimmtes Maß, wird nochmals rekursiv aufgeteilt und wieder der Fehler gemessen. Die exakte Oberfläche wird per trilinear Interpolation bestimmt.

## Glättung

Zur Behebung von Artefakten die bei der Erstellung von Oberflächenrepräsentation (besonders bei sehr anisotropen Daten) entstehen. **Input:** Noise in den Objektgrenzen, kleine losgelöste Objektteile, Löcher. (Führen besonders stark zu ungewollten Artefakten)



### Charakteristika:

1. Reduzierung von HF-Noise bei Erhaltung wichtiger Features
2. Geschwindigkeit / Performanz des Algorithmus
3. Genauigkeit der Ergebnisse

### Messungen:

1. Distanz zwischen geglätteter und originaler Oberfläche
2. Erhaltung des Volumens
3. „Curvature plots“

### Vergleich

1. Anzahl an Iterationen
2. Gewichtungsfaktoren
3. Nachbarschaften
4. Qualität
5. Volumenserhaltung
6. Laufzeit

Glättung von Oberflächen kann auch als Optimierungsaufgabe gesehen werden: Minimierung von totaler Krümmung und Oberfläche.

## Morphologische Operationen

### Modifizierte Erosion

$$v = v_2 - (v_2 - v_1) \cdot \frac{1}{3}$$

### Modifizierte Doppelte Dilation

$$v = v_{ref} - (v_2 - v_1) \cdot \frac{d}{3}$$

### Laplace'sche Glättung

Betrachte Vertices  $q_i$  innerhalb topologischer Distanz 1 von Vertex  $p$ . Die neue Position von  $p$  die als  $p'$  bezeichnet wird, ergibt sich folgendermaßen:

$$p' = p + \frac{\lambda}{n} \sum_{i=0}^{n-1} (q_i - p)$$

Wobei  $\lambda$  den Glättungsfaktor und  $n$  die Anzahl an Durchgänge beschreibt. Laplace'sche Glättung ist simpel und schnell.  **$p'$  bewegt sich Iteration für Iteration zum „Zentrum“ seiner Nachbarn.**

**Problem:** Ungewolltes schrumpfen, kleine Features werden schnell weggeglättet. **Lösung:** Vertices werden eine **bestimmte Distanz pro Iteration zurückbewegt**. Weitere Parameter: Wie weit zurückbewegen und wie viele Nachbarn sollte man einbeziehen?

### Tiefpassfilter

Anwendung zweier Laplace-ähnlicher Glättungsoperationen, die sich durch Glättungsfaktor unterscheiden. Die Nachbarn haben jedoch alle gleichen Einfluss.

$$p' = p + \frac{\lambda}{n} \sum_{i=0}^{n-1} (q_i - p)$$

$$p' = p + \frac{\mu}{n} \sum_{i=0}^{n-1} (q_i - p)$$

Trotz Korrekturen gehen bei Tiefpass-Filtern kleine Details leicht verloren. Tiefpass ist **generell eine gute Wahl**. (alle Objekte)

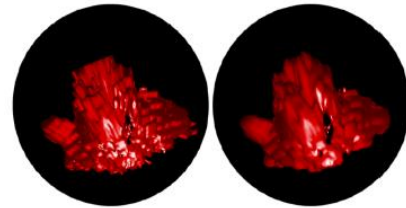
### Weitere Herangehensweisen

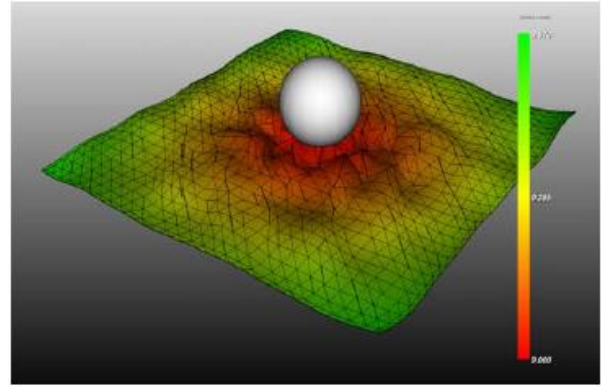
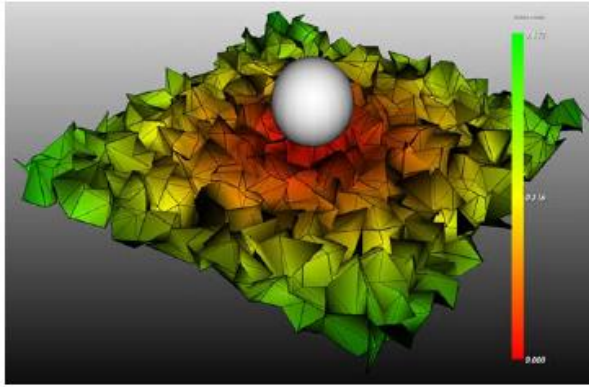
1. **Mean curvature Flow:** Vertices werden nach Objektkurvatur modifiziert
2. **Average Filtering:** Normale werden durch Durchschnitt der Nachbarschaft ersetzt
3. **Median Filtering:** Normalen der Nachbarschafts Voxel sortiert nach Winkel des derzeitigen Voxels

### Distance-Aware Smoothing

Es wird geglättet unter Berücksichtigung klinisch relevanter Aspekte wie unerwünschte Volumesreduktionen (beu Tumoren etc.), Formveränderungen und Distanz zwischen Strukturen.

0	0	0	0	0	0	0
0	0	0	100	100	100	0
0	100	100	100	100	100	0
0	100	100	100	100	100	0
0	0	0	100	100	100	0
0	0	0	0	0	0	0

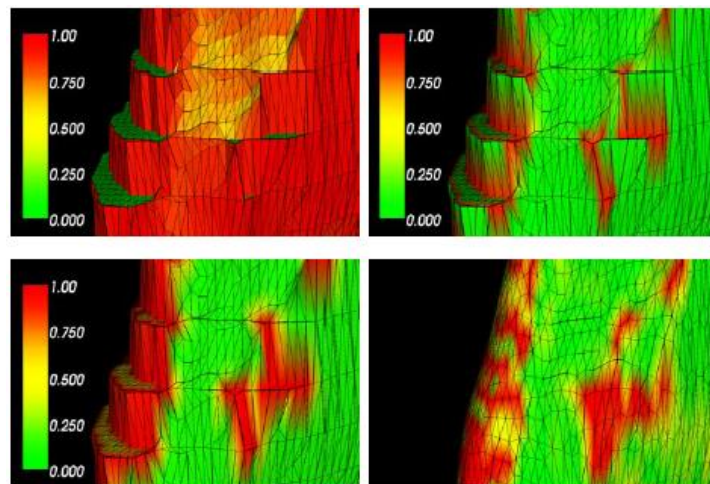




**Probleme:** Entfernt möglicherweise immernoch relevante Details (nicht perfekt), Staircase-Artefakte

### Staircase-Aware Smoothing

Abhängig von **Orientierung der Bilddaten**, **Inter-Slice-Distanz** tauchen scharfe Kanten auf (=Staircase-Artefakte).



1. **Face-Orientierung** feststellen
2. **Änderungen in Face-Orientierung** feststellen
3. **Gewichte nach Distanz der Vertices zu den Treppenecken**
4. Führe **distanzgewichtetes Smoothing** durch

**Parameter:** Weite/Distanz von Filter, Staircase-Detektions-Threshold.

## Kapitel 4: Direct Volume Visualization

Der Versuch die Information vieler Slices „unverändert“ und direkt darzustellen. Basics davon sind: Orthogonale oder Oblique Slices, Curved Planar Reformations (CPR) und Multiplanare Reformations (MPR)

### Grundlagen

#### Anwendungen:

1. **Anatomie:** Ausbildung und Untersuchungen
2. **Radiologie:** Blutgefäß-Diagnosen und Läsionbewertungen (z.B. bei Tumoren)
3. **Therapie Planung:** Operations Planung und Simulation, Radiationsdosis

#### Anforderungen:

1. Detaillierte visuelle Repräsentation von Originaldaten (Wichtig für korrekte Diagnose und Therapie)
2. Gute visuelle Repräsentation räumlicher Beziehungen (auch visuell gestützt per Highlights und Schatten)
3. Schnelles Rendern
4. Unterstützung unterschiedlichster Datentypen (Ach ja?)

Voxel (Volumselement)/Zelle, was das ist ist klar. Ein **getaggttes Volumen** bezeichnet einen Volumensdatensatz wo alle Voxel zu einem segmentierten Objekt gehören. Wenn die TF allen Voxel Opazitätswerte verliehen hat, kann man es als **klassifiziertes Volumen** bezeichnen.

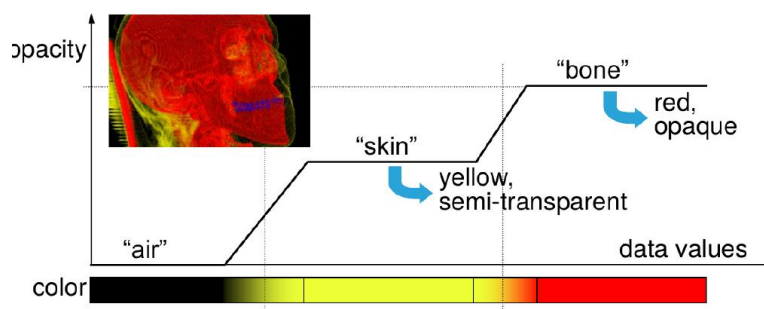
Ein medizinischer Datensatz besteht meist aus **Voxel in einem regulären orthogonalen Grid**. Die Daten sind **meist anisotrop** (Slice-Distanz nicht äquidistant), typischerweise CT oder MRI Daten mit (512x512x100-2000), wenn hochaufgelöst (1024x1024x100-2500). **12 bits pro Voxel bzw. 16 bit unsigned/signed short**. PET, SPECT und 3D-US hat eine geringere Auflösung.

Der **Gradient** ist ein **Vektor der in die Richtung der größten Veränderung** zeigt. Volumsdaten können als Funktion von R3 in R1 gesehen werden. Der Gradient einer Funktion entspricht der ersten Ableitung. Die Länge des Gradienten ist ein gutes Maß für Änderung und eignet sich zum Einsatz für „**Opacity Modulation**“. Möglichkeiten den Gradienten zu berechnen:

1. Central Difference  $\frac{1}{2}(+1 - -1) \dots f(x+1) - f(x-1), f(y+1) - f(y-1), \dots$
2. Forward Difference  $(+1 - 1)$
3. Backward Difference  $(1 - -1)$
4. Sobel Kanten Detektor

## Transferfunktion

**Transferfunktionen mappen Intensitätswerte des Volumens auf Farbe und Intensität.** Bei einem gut gemachten Transferfunktionswidget kann der Benutzer in Anbetracht des Histogramms Regionen abgrenzen, welchen dann bestimmte Werte zugeordnet werden.



Gute Transferfunktionen zu definieren ist keine leichte Aufgabe, sie haben jedoch maßgeblichen Einfluss auf die endgültige Darstellungsqualität. Auch wenn es sogar 2D und 3D Transferfunktionen gibt, sind die Segmentationsmöglichkeiten begrenzt.

TFs für Blutgefäße sind mühsam zu definieren, da zu deren Darstellung oft non-uniform verteilte Kontrastmittel eingesetzt werden. Um sie homogen darzustellen ist es erforderlich die TFs fein zu adjustieren.

## Volume Rendering Integral und Ray-Casting

**Image-based Verfahren:** Für jeden Pixel im Bild wird ein Strahl in die Szene geschickt und Schnittpunkte mit dem Objekt berechnet. Farbe und Opazität wird aufsummiert während der Strahl das Objekt durchwandert.



Bei **Object-based** Verfahren läuft's genau anders rum: Es wird **zu jedem Objekt-Voxel** berechnet, was er zum endgültigen Bild beiträgt. (z.B. per Splatting)

**Texture-based** Verfahren nutzen die Möglichkeiten **moderner 3D Hardware** aus. Laufen über Shader (GLSL, HLSL, CUDA, ...)

$$I_{\lambda}(x, r) = \int_0^L C_{\lambda}(s) \cdot e^{-\int_0^s \alpha(t) dt} ds$$

$I_{\lambda}$  amount of light with wavelength  $\lambda$

$x$  image plane position

$r$  direction of ray

$L$  length of ray

$s$  current sample along ray

$C_{\lambda}$  light of wavelength  $\lambda$  reflected or emitted from  $s$  in direction  $r$

$\alpha$  density of particles reflecting light to viewer

$$I_{\lambda}(x, r) = \sum_{i=0}^{L/\Delta s} \left[ C_{\lambda}(s_i) \cdot \prod_{j=0}^{i-1} (1 - \alpha(s_j)) \right]$$

$C_{\lambda}(s_i)$  color at position  $s_i$

$\alpha(s_i)$  opacity at position  $s_i$

$\Delta s$  step size

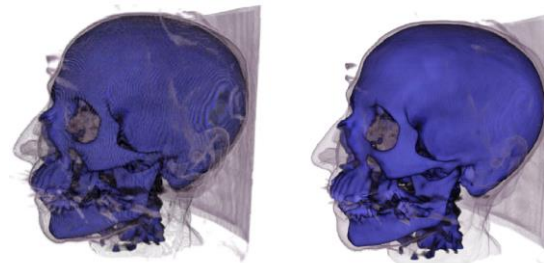
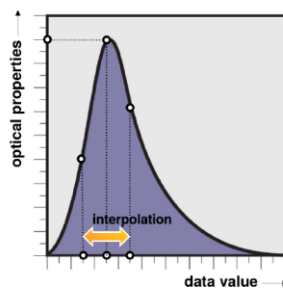
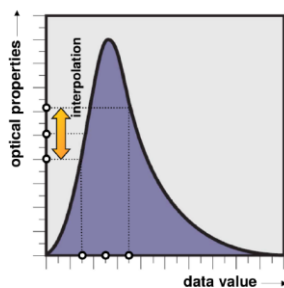
**Volume Rendering Integral:** Links exakt, Rechts numerisch approximiert

**Klassifikation:** Anwendung von TF um Opazitätswert  $\alpha$  zu erhalten. Phong Shading per Gradient.

**Shading:** Anwendung von TF um Farbe  $C$  zu erlangen. Einsatz von Gradient zur Kantenbetonung. (Materialübergänge)

■ **Pre-interpolative:** Classify data values (apply TF) → interpolate color (RGBA) values

■ **Post-interpolative:** Interpolate data values → classify result (apply TF)



Left: Pre-interpolative. Right: Post-interpolative. (Images: Eduard Gröller)

## Techniken

### Projektionstechniken

**Einzelne Datenwerte aus Set werden (exakt, selektiert) dargestellt**, insofern werden auch nur unveränderte Originalwerte eingesetzt. Transferfunktion muss auch nur optionalerweise definiert werden. Techniken gut akzeptiert, Ärzte mögen es wenn die Werte unverändert sind. Jedoch nehmen diese Techniken die **räumliche Komponente weg**, es kommt zu Mängeln bei der Tiefenwahrnehmung und Okklusion.

Technik	Funktion	Eigenschaften
First Hit	Nimm den ersten Wert auf den der Ray trifft	Ordnungsabhängig
MIP	Nimm den maximalen Wert auf den der Ray (bis zur Termination) trifft	Ordnungsabhängig, Noisedämpfend, kann zur Visualisierung von Blutgefäßen eingesetzt werden

LMIP	Nimm den ersten Treffer der über einem bestimmten Threshold liegt	Ordnungsabhängig, Blutgefäße, erhält Gefäßgrenzen und räumliche Abhängigkeiten
MinIP	Nimm den minimalen Wert auf den der Ray (bis zur Termination) trifft [ignoriere dabei Hintergrund] (CFA = zirkuläre (nicht lineare) Rays)	Ordnungsabhängig, kann Soft-Plaqua in CTA aufzeigen, Nützlich im Stenose zu graden

### Kompositstechniken

Die Werte werden entlang dem Viewing Ray akkumuliert. Räumliche Information wird hierbei einbezogen, was zur sehr realistischen und intuitiven Darstellungen führt. Daten werden jedoch „verändert“ und die Berechnungszeiten liegen höher.

#### ■ Front-to-back (under operator)

$$C^{out} = C^{in} + (1 - \alpha^{in}) \cdot C$$

$$\alpha^{out} = \alpha^{in} + (1 - \alpha^{in}) \cdot \alpha$$

#### ■ Back-to-front (over operator)

$$C^{out} = C + (1 - \alpha) \cdot C^{in}$$

$C^{out}$  accumulated color  
 $C$  sample color  
 $\alpha^{out}$  accumulated opacity  
 $\alpha$  sample opacity

Technik	Funktion	Eigenschaften
Average	Mittel entlang Ray	Liefert simuliertes Röntgenbild
MIDA (Maximum Intensity Difference Accumulation)	<p>Benötigt keine TF, Jeder Veränderung der Intensität entlang des Viewing Rays wird „klassifiziert“. (Immer wenn sich Intensität ändert und die neue Intensität ist größer als die derzeit bekannte maximal Intensität <math>I_{max}</math> wird Opazität für den derzeitigen Sempel erhöht = Beitrag zur akkumulierten Intensität des Rays)</p> $\delta = \begin{cases} I - I_{max} & \text{if } I > I_{max} \\ 0 & \text{otherwise} \end{cases}$ <p>Weighting factor with DVR (-1) → MIDA (0) → MIP (+1) transition parameter <math>\gamma</math></p> $\beta = \begin{cases} 1 - \delta \cdot (1 + \gamma) & \text{if } \gamma < 0 \\ 1 - \delta & \text{otherwise} \end{cases}$ <p>Compositing</p> $C^{out} = \beta C^{in} + (1 - \beta \alpha^{in}) \cdot \alpha \cdot C$ $\alpha^{out} = \beta \alpha^{in} + (1 - \beta \alpha^{in}) \cdot \alpha$	MIDA Bild
DVR	Opazität und Farbe aus TF	DVR Rendering unter Einbezug der TF



## Ray-Casting Eigenschaften

### Beschleunigungstechniken

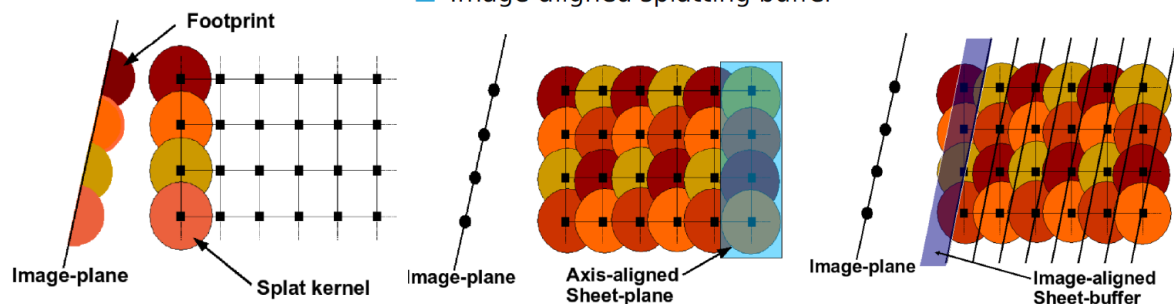
1. Leere Zellen skippen
2. Early Ray termination
3. Adaptive Refinement
4. Template-based Viewing
5. Uvm.

### Performanz Aspekte

1. Projektionstyp
2. Viewing-Richtung
3. Grayscale oder Farbe?
4. Eingesetzte Beschleunigungstechniken und Shortcuts
5. Festplatten und RAM-Leistung
6. RAM und Cache gröÙe
7. Anzahl von Rays und Voxeln

## Objekt-Basierte Techniken

- Axis-aligned splatting buffer
  - ▶ Splatting traversal order depends on viewing direction
  - ▶ Most parallel projection to viewing direction chosen
  - ▶ Popping artifacts due to view changes
- Image-aligned splatting buffer



## Hybridtechniken

**Kombination** von Surface(**Indirect**)-Volumensvisualisierungen und **direkten** Volumensvisualisierungen. **Möglichkeiten**

1. Surface Visualisierungen segmentierter Objekte integrieren in original Daten (z.B. BlutgefäÙe)
2. Kombination von Volumensdaten mit zusätzlichen graphischen Elementen
3. Einbezug geometrische Modelle von Werkzeugen wie Skalpell

### Herangehensweisen

1. Render Surface per z-buffer, rendere Volumsdaten (con: Transparente Oberflächen)
2. Voxelization von Surface, Rendern der ganzen Szene in DVR
3. Aufteilung von Volumen in geslicte Textur-Dreiecke, sortiere Triangel nach Tiefe

# Kapitel 5: Interaktionstechniken

## Erkundung und Reduktion

Im simpelsten Fall: Einfach durch die Schnitte scrollen, vll noch Helligkeit/Kontrast adjustieren und Metadaten anzeigen. **Weitere Möglichkeiten:**

1. **Simultane Anzeige mehrerer Datensets** (CT von Organ mit und ohne Kontrast)
2. **Slice/Schnittflächen Orientierung frei wählbar** (Orthogonal, Oblique, Multi/Curved Planar)
3. **Fensterfunktion** (Weite und Zentrum, Helligkeit und Kontrast)
4. **Matrix-Ansicht unterschiedlicher Slices**

### Reduktion allgemein

Daten werden wegen einer Vielzahl an **Gründen** reduziert: Um Noise zu entfernen, um den Fokus auf bestimmte Regionen der Daten zu legen oder auch wegen Memory bzw. Performance Constraints.

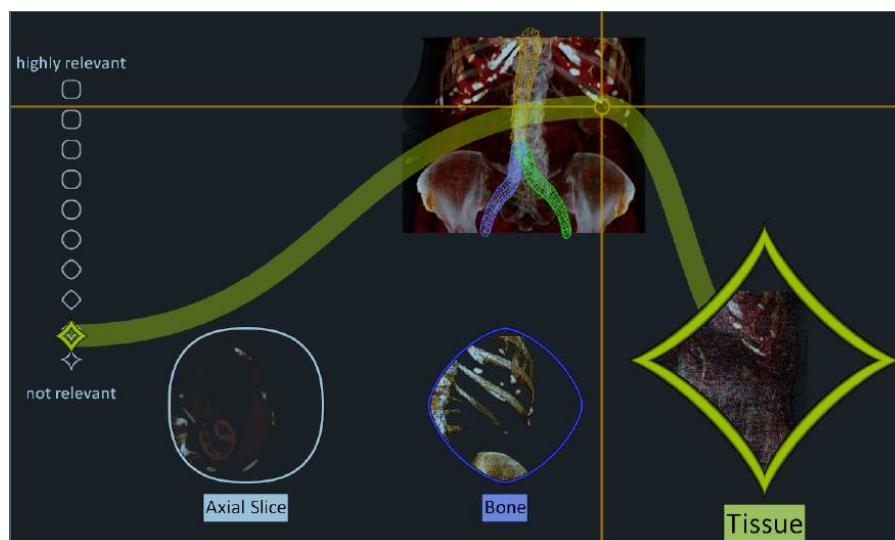
### Möglichkeiten:

1. **Selektion eines Datenintervalls**
2. **Downsampling**
3. **Definiton von Regions of Interest**
4. **Projektion** (First Hit, MIP, ...)
5. **Datenaggregation**

Man versteckt irrelevante Daten um den Fokus auf die relevanten zu lenken.

### Smart Super Views

Technik für Systeme mit vielen unterschiedlichen Views und Visualisierungstechniken. Es gilt die Entscheidung zu treffen **welche View für welche Situation** am geeignetsten ist. Bei Smart Super Views werden aufgrund von Parametern wie **Modalität der Daten**, **Generierte Daten** und **weiterem Vorwissen** ein „Ranking“ an Views erstellt, die dem Benutzer zum Einsatz (Relevanz kodiert durch Form) suggeriert werden.



# Transferfunktionen

Bereits bekannt. **Challenges:** Korrekte Ausprägung nicht straight-forward, teils schwer zu finden, da gutes TF-Design Vorwissen und Erfahrung erfordert. Viele Parameter erhöhen Flexibilität aber auch Komplexität → macht Design von TF noch schwerer. **Möglichkeiten dem entgegen zu kommen:**

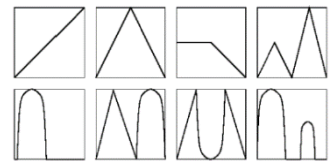
1. **Vordefinierte TFs** anbieten (mit intuitiver Suche)
2. **Zusammenhang zwischen Parametern und Ergebnis** sollte gut erkennbar sein
3. Schnelles **TF Preview**

Transferfunktionen müssen nicht zwingend wie bei DVR sein. Windowing, und Verwandte (Bi-Window, Inverse-Window) sind auch „Transferfunktionen“.

Transferfunktionen: Stochastische Generation

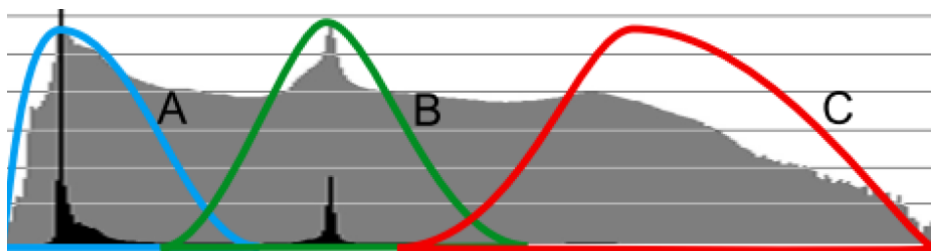
**Automatische Generation** von Transferfunktion auf Basis **genetischer Algorithmen**. Ablauf:

1. Start mit einem Satz an **initialen TFs**
2. **Mutation** der TFs mit einem genetischen Algorithmus (25 Generationen)
3. DVR → **Subjektive Evaluation** durch Benutzer
4. Zurück zu 2



Transferfunktion: Image-Based

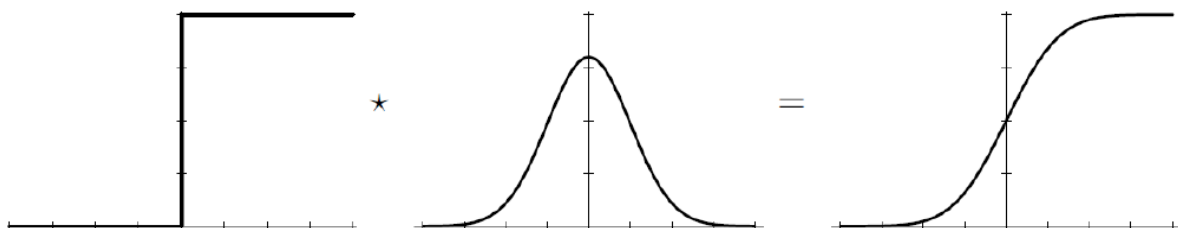
**Bildinformationen werden genutzt** um TF zu definieren. z.B. per globalen Histogramm oder Slice-Histogramm.



Auch Möglich: Intensitäts-Profil entlang Viewing Ray.

Transferfunktion: Data-Based

Eine Möglichkeit für datenbasierende TFs sind die Kanten-betonenden TFs nach Kindlmann und Durkin. Annahme für diese Technik: Die **gewollten idealen Kanten** sind aufgrund von Bildfehlern **gauss'sch Verunschärft**. Die Form der optimalen Kanten ergibt sich dadurch aus der **Kombination mit einer Gauss-Glocke**. Die Ecken werden somit nicht im räumlichen Domain, sondern direkt in den Daten gesucht.

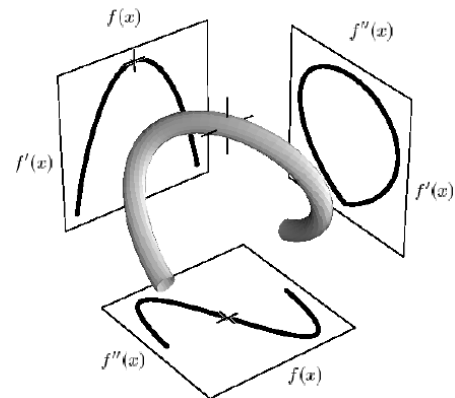


Weiters haben Kanten einen hohen Gradient  $g$  und die 2. Ableitung an sollte möglichst gering (nahezu 0) sein. **Intern** wird bei dieser Technik ein **3D-Histogramm** erstellt mit  $X: f(x)$ ,  $Y: f'(x)$ , und  $Z: f''(x)$ . Der

Algorithmus berechnet nun für den **ganzen Datensatz** die **min** und **max-Werte** für  $f''(x)$  und  $f'(x)$  (min  $f'(x) = 0$ ) und **befüllt H** wo **min** und **max** von  $f''(x)$  und  $f'(x)$  einen Wert zwischen **[0,255]** **gemappt** **kriegen**. (Hat was mit Bins zu tun, ich checks NOCH nicht. :D)

Siehe hier: <http://people.cs.uchicago.edu/~glk/pubs/pdf/Kindlmann-SemiAutomaticTransferFunctions-VV-1998.pdf>

Das **Fadenkreuz** beim 3D-Histogramm (rechts) stellt die **Mitte der Grenze** dar. Das **Histogramm Volume H** kann dann die **Position von Kanten in Relation zu den Daten** darstellen. Der Benutzer kann sich dann Peaks (möglicher Ausprägungen der TF) für TF anzeigen lassen, und diese auch in ihrer Form anpassen. Typisch sind Dreiecks- und Quadratformen.



Diese Technik funktioniert recht gut, **erfordert** habe **saubere Objekt-Grenzen** und **homogene Daten**. Es darf auch **nicht zu Noisy** sein und sollte möglichst **keine Outliers** haben.

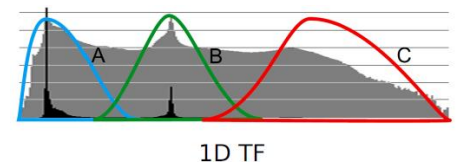
Die **empirisch entdeckte TF** kann auch **wiederbenutzt** werden. Z.B. mehre Aufnahmen Aneurismen bei selber Modalität (MRI) **Hierzu:**

1. **Referenzdaten Dref** und **Referenzfunktion Tfref** auswählen
2. **Normalisiere Histogramm** von Datensets **H(Dref)** und **H(Dstudy)**
3. Non-Lineare **Transformation t** von den Intensitätswerten von **Dstudy**, sodass **H(Dstudy) ~ H(t(Dref))**
4. Dann gilt **Tstudy(v) = Tref(v)**

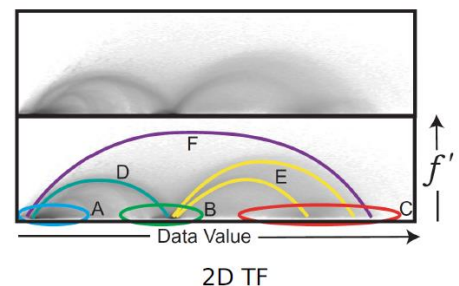
Die Ähnlichkeit der Histogramme kann relativ einfach über die Summierte Distanz von  $H(v)$  bestimmt werden.

Transferfunktion: Multi-Dimensional

1D nutzen nur Intensität, **nD benutzen weitere Werte wie Gradient, 2. Ableitung** etc. (Gradient Opazitätsmodulation) Das erlaubt bessere Resultate, macht aber die Definition von der TF wiederum komplexer.



Die 1. Ableitung ist ein Vektor, die 2. ist eine Matrix und zwar die Hesse Matrix. Bei der 1. Ableitung kriegt man den **Skalarwert** einfach aus der Vektorlänge. (Gradientmagnitue) Bei der 2. Ableitung Ist die Matrix eine „**Hesse Matrix**“, der **Skalar ergibt sich aus dem größten Eigenwert der Matrix**.



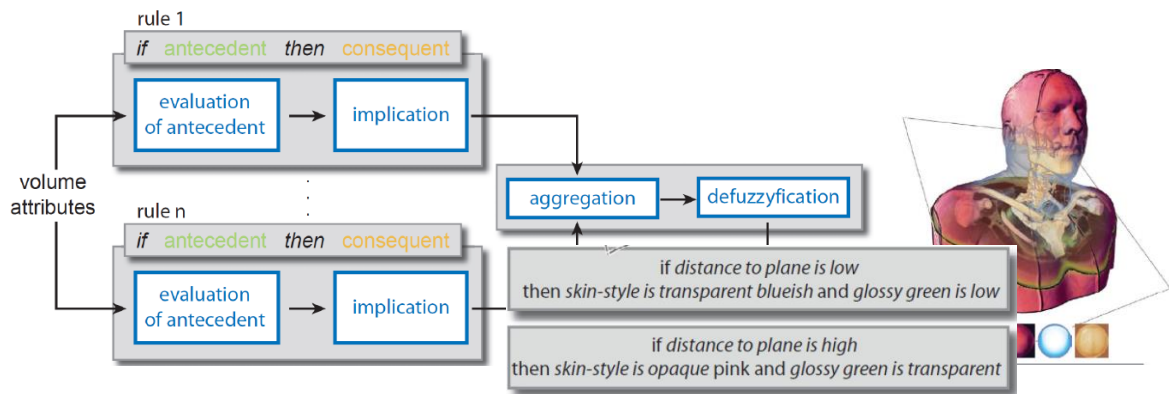
Transferfunktion: Lokal

Einfache Idee: Anstatt TF global zu definieren, gelten **unterschiedliche TFs für unterschiedliche Regionen** des Bildes. Erlaubt bessere Gewebsunterscheidung.

## Weitere Selektionstechniken

Semantic Visualization Mapping

Mappt Semantik zu unterschiedlichen Visualisierungen, Abhängig von Viewpoint/Benutzerfokus (**Interaktion**), Modalität/Protokoll (**Daten**) und Rendering Technik. (**Illustration**). Benutzt Fuzzy Logic zur Regelevaluation. (Ich denke die Smart Super Views gehören hier übrigens auch dazu, denke ich)



## Volumetrische Selektion

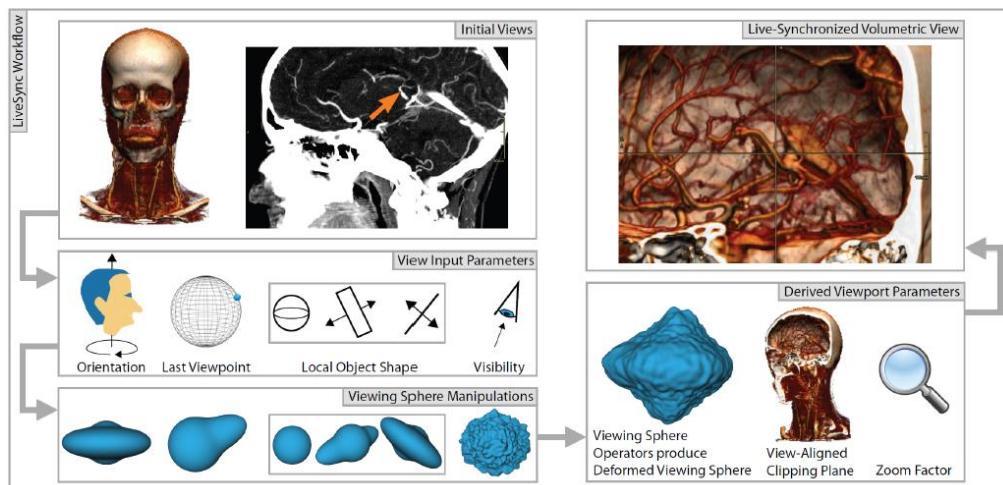
**Selektion eines Volumens innerhalb eine Szene verschiedener Objekte.** Sind alle Objekte undurchsichtig, einfach das „erste“ bzw. vorderste auf das der Benutzer klickt. Bei Transparenz wird's schwierig.

- Möglichkeit 1: **Dämpfe „Klick-Ray“ nach Größe und Opazität** der Objekte die er trifft. **Kleinere und undurchsichtige Objekte werden eher gewählt** als große Transparent. Probleme bei semi-transparenten Objekten kleinerer Größe die sich im Vordergrund befinden.
- Möglichkeit 2: **Objekte entlang „Klick-Ray“** werden User **präsentiert**, er kann dann einfach **durchscrollen**.

## LiveSync

Eine 2D und eine 3D View. Ein Klick auf ein Slice der 2D View selektiert das Volumen in der 3D View. Damit das klappt muss der Picking-Algorithmus Dinge wie **Patienten Orientierung, Viewpoint History, Object Shape** und **Sichtbarkeit** berücksichtigen.

LiveSync versucht das Problem zu beheben, dass das Einstellen mehrere Views sehr mühsam ist. (2D und 3D). Man klickt eben z.B. einfach auf eine Region in einem 2D Slice und alle anderen 2D und 3D Views werden automatisch adjustiert um gute Sicht auf die Region zu geben.



Die Verbesserung LiveSync++ zeigt **selektierte Objekte „unobstruiert“**, also frei. Das Funktioniert per **automatischer Readjustierung der TF** basierend auf der lokalen Intensitätsverteilung in der Nähe der gewählten Region.



## Contextual Picking

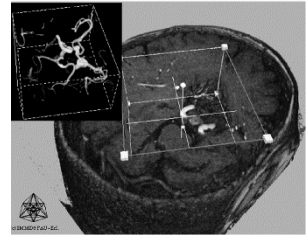
Ähnlich läuft Contextual Picking, man **wählt etwas in der 3D Visualisierung aus** und der Picking Ray wird so analysiert, dass die besten/geeignesten **2D Views angezeigt** und ein „**interessanter Punkt**“ in diesen Views **gehighlighted** wird.

## Contextual Snapshots

Erlaubt die interaktive Definition räumlicher Annotationen. Selektionen werden Kontextabhängig gespeichert (Orientierung, Transparenz, ...) Über die Annotation sind somit unterschiedliche Views in unterschiedlichen Kontexten anwählbar.

## Erkundung per Local Volume Rendering

Es wird eine lokale Clipping-Box samt lokaler TF definiert. Fokussierung auf kleinere Region erlaubt mehr Details. (Tumorstruktur besser in 3D erkennbar, Segmentierung einfach in kleinen Sub-Volumen)

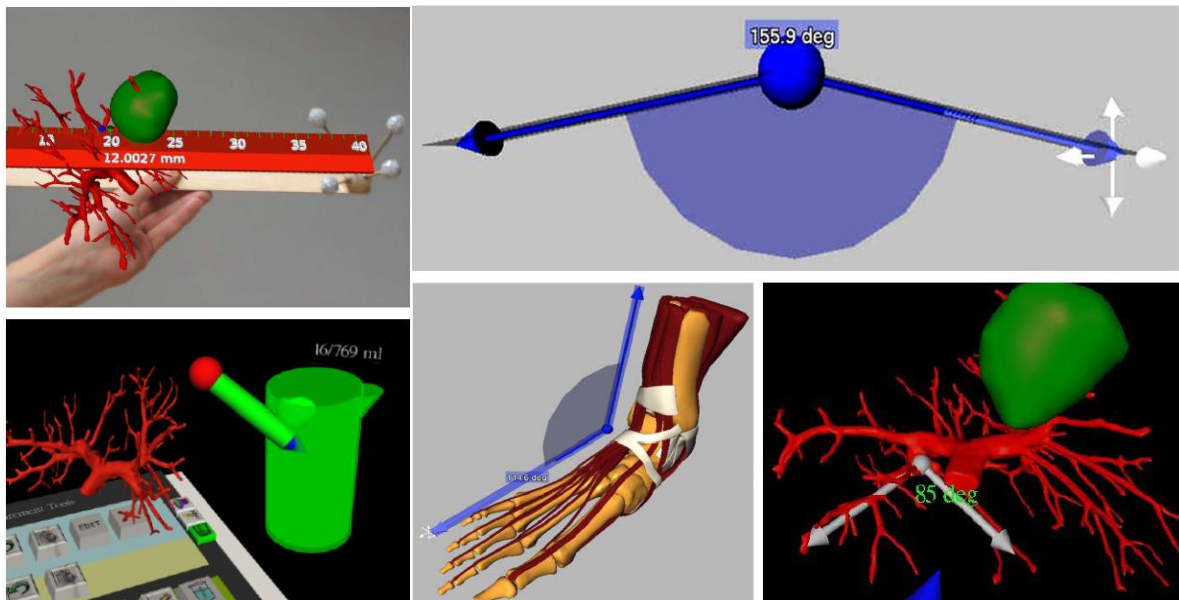


## Messungen

Notwendig für Diagnosen, Rating/Staging von Krankheiten, Überprüfung von Therapieerfolgen und Therapieplanung.

1. **Durchschnittlicher Grauwert innerhalb einer Region** (z.B. bei Osteoporose)
2. **Winkel** (Gelenkfehlposition?)
3. **Distanz** (Tumor zu Gefäß)
4. **Diameter** (Blutgefäßrekonstruktion)
5. **Volumen** (Größe von Tumoren)

In 2D ist das ganze simpler, jeder Voxel ist auswählbar. Abgesehen von Winkeln von eigentlich 3D-Objekten. Das lässt sich nur schwer approximieren.



## Pfadplanung

Biopsie Planer: Sicherster Weg für Biospy von (Gehirn)tumoren? Pfad anzeigen. Übersicht in 3D, Details in 2D. Unterstützung per Histogramm. Distanz zu Vitalstrukturen sollte angegeben werden. Sicherheitsabstand und Pfadanalyse. 3D DVR und 2D Slice-View.

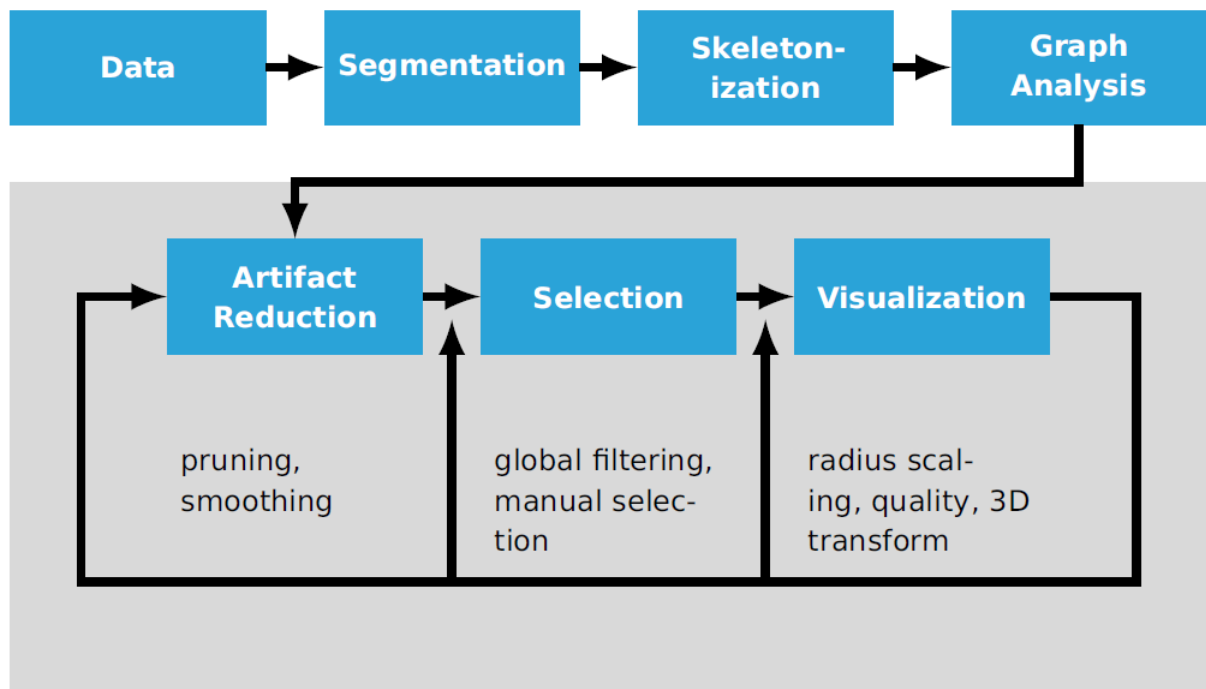
## Kapitel 6: Blutgefäß Visualisierung

Anwendung: Zur Diagnose von Gefäßerkrankungen und Behandlungs/OP-Planung. Erkennbare Läsionen und Krankheiten: Stenosen, Okklusion und Aneurismen. Die Daten sind (meist/nahezu immer) Angiographie-Daten.

### Konventionelle Methoden:

1. **MIP/DVR**
2. **Shaded Surface Display (SSD)**
3. **Verbesserungen durch CCA** (beseitigt lose Fragmente)
4. **Exploration:** Visualisierung komplexer Gefäßsysteme, Variation in Transparenz und Radius. Probleme ergeben sich als **Mehrdeutigkeiten bei Bild-basierten Selektionen**. (z.B. Volumenselektion) Auch hierarchische Selektionen (Unterbäume) schwer selektierbar.

### Gefäßvisualisierungs Pipeline



#### Data

Gefäße werden rausgefiltert, indem man versucht **sehr gestreckte Objekte** zu finden. Und zwar per Eigenvalues der Hesse Matrix.  $\lambda_1 \gg \lambda_2, \lambda_3$  **Problem:** Zerfall in Teilstücke (da wo CCA hilft)

**Multi-Scale-Analysis** ist ebenfalls eine Technik um die Daten aufzubereiten. Hierbei werden die Gefäß(ähnlichen)objekte in **verschiedenen Größen** innerhalb der Daten verbessert. Die Zwischenergebnisse werden sodann in einen **einzelnen Datensatz kombiniert**. Maximum Response (Gefäßgröße) wird gespeichert. Dann werden noch die Zentrumslinien der Gefäße berechnet und kleinere Subgefäße erfasst.

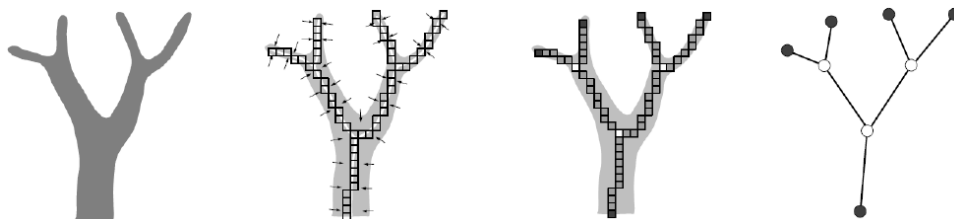


### Segmentierung

Hier wird ein **Hess'sches Thresholding** eingesetzt um das **Volumen zu verbessern**, **Hysteresis Thresholding** kann eingesetzt werden um **kleine Objekte zu entfernen**. Die **gethresholdeten Volumen** werden dann letztendlich zu einem **kombiniert**, die Maximum Response in ein separates Volumen gespeichert. **Berechnung des Skeletts der kombinierten Volumen** führt zu den **Gefäß-Zentrallinien**.

### Skeletonization

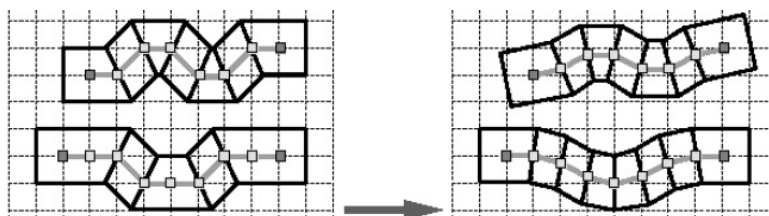
Mit bereits bekannten Verfahren zur Verdünnung. (u.a. Medial-Achsen Transform, Reeb Graphen)  
**Ergebnis ist eine Graph-Präsentation der Gefäße**



**Graph-Analyse:** Der Gefäß-Graph/Baum hat dann **Braching-Points und End-Points**. Ein Teilstück das **zwischen 2 solcher Punkte** liegt wird als **Segment** bezeichnet. Diese Technik ist allerdings anfällig für Noise in den Gefäßgrenzen. Es kann zu ungewollten kleinen Side-Banches kommen.

### Artefakt Reduktion

**Unebenheiten** in den Segmenten können relativ „einfach“ **per Smoothing entfernt** werden.



Die bereits erwähnten kleinen **Sidebranches** können durch Einbeziehung des **Main-Branch/Side-Branch Ratios** erkannt werden. Weitere Möglichkeiten sind die **Ablehnung von Deformationen** und das **Zentrieren von Voxeln bei symmetrischen Bifurkationen**.

**Generell:** **Segmente** gehören **verbunden**, **Kleine und kurze Branches entfernt** und **Unebenheiten** bzw. „Löcher“ sollen **minimiert** werden.



## Selektion

(Vermutlich für Selektion) **Berechnung von Gefäßcharakteristika** (Dicke, Länge, vermutlich um zusammengehörende Segmente erkennen zu können). Erlaubt **Filtern und Erkunden** des Gefäßbaumes.

## Visualisierung von Gefäßen: Modellbasiert

Soll räumliche Position, Branching-Struktur und Radii von Gefäßen beschreiben. Es wird versucht das Gefäßsystem als gerichteter Graph mit Branching-Points als Vertices darzustellen. Die Gefäße sind entweder zirkulär oder elliptisch.

### Truncated Cones

Die **Gefäßsegmente** werden **als abgeschnittene Kegel** dargestellt. Die **abgehackten Endflächen werden durch Sphären ersetzt**. Diese Darstellungsform ist eine einfache und schnelle. Probleme entstehen durch Unebenheiten bzw. Lücken die wegen der Orientierung bzw. den unterschiedlichen Radii der Kegel entstehen können.

Um den Lücken wegen unterschiedlicher Radii entgegenzuwirken kann man **Radius Smoothing** einsetzen. (Größe von Radius wird durch Einbezug von Nachbarschaft neu gesetzt / gemittelt.)

Bei Objekten mit **sehr starker Krümmung** ist **Refinement** notwendig.

Weitere Probleme:

1. **Radii sind nicht immer akkurat / verlässlich**
2. **Dünne Gefäße** sind **problematisch** (nur 1-2 Voxel teils dick)
3. **Pathologische Veränderungen der Gefäße** führen zu **inhomogenitäten**

### Vergleich modellbasierter Herangehensweisen

	<b>Symbolic Rep.</b>	<b>Truncated Cones</b>	<b>SSD</b>	<b>DVR/MIP</b>
Abstraction degree	High (radius ignored)	Medium (circular vessel model)	Low	Very low
Memory consumption	Low ( $\approx 10^2 \dots 10^3$ coordinates)	Medium ( $\approx 10^3 \dots 10^4$ coordinates & radii)	High ( $\approx 10^5$ triangles)	Very high ( $\approx 10^6$ voxels)
Branching structure	Optimal	High	Problematic	Very problematic
Interaction (filtering, highlighting)	Possible, regarding hierarchy	Possible, regarding hierarchy & radii	Problematic, clipping planes required	Problematic, clipping planes required

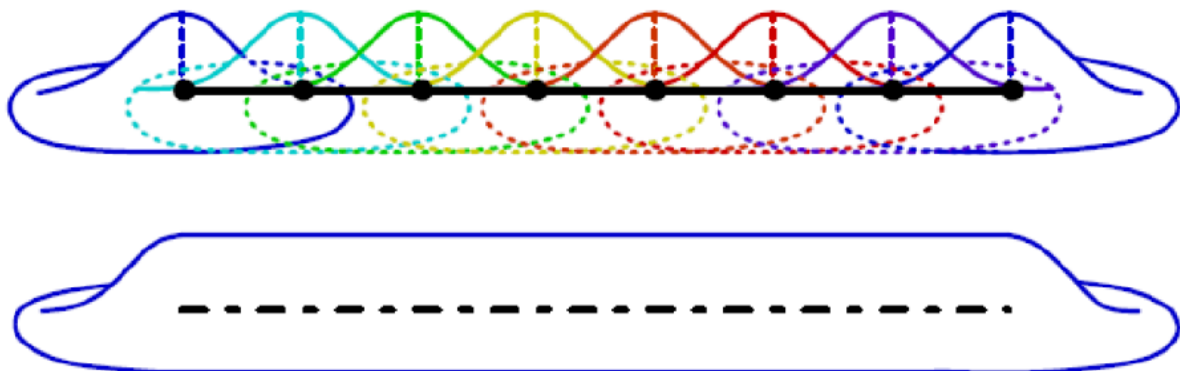
### Implizite Methoden

Basiert auf Impliziten Oberflächen. Es wird eine **implizite Funktion** definiert, die bei bestimmten Werten immer 0 liefert. ( $F(x, y, z) - Iso = 0$ )  $F()$  kann auch als „skalare Feldfunktion“ bezeichnet werden, denn sie berechnet für jeden Punkt  $(x, y, z)$  einen Skalarwert. Der **Iso-wert generiert Isosurfaces**, welche die Oberfläche beschreibt, wo die Implizite Funktion 0 ist.

Anstatt also Cones etc. zu auf das Skelett zu mappen wird einfach eine Oberfläche per impliziter Funktion relativ zum Skelett definiert. **Convolution Surfaces** verhindern Wülste und Falten. Bei der Erstellung des Convolution Surfaces werden alle Punkte des Skeletts integriert

Um smooth Oberflächen auf den Linien-Segmenten zu kreieren: **Convolution Surfaces**.

1. **Signal wird mit einem Filter gefaltet** (Signal = Skelett)
2. **Linien-Segmente** werden **Tiefpass** gefiltert (Hohe Frequenzen fallen ab, ein smoothes Feld wird erstellt = implizite Isosurface)
3. **Polygonisation per impliziter Funktion mit Iso-Value** (abhängig von angewandten Filter)
4. **Marching Cubes** kommt zum Einsatz



- **Pro:** Smooth Oberflächen, alles sieht sehr organisch aus
- **Con:** Branch merging

**Problemfälle:**

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1413501>

**Problems:**

- **Bulging:** Interfering scalar field at bifurcations  
→ Undesired thickening



(Oeltze and Preim 2005)

- **Blending strength:** Merging of neighboring segments



Bluging und Blending gehören in der Grafik vertauscht. Problem kann durch Verwendung eines „**small Gaussian**“ bzw. „**narrow Gaussian**“ Filters aufgehoben werden.

Die **Berechnung der Skalarfelder** ist **aufwendig**. Verbesserung: Erstellung von **Bounding Boxes**. Berechnung nur innerhalb von Grenzen.

Size-Based und räumlich variierende TFs

Größe wird durch TF kodiert. Berechnung von Größe durch multi-scale Analysen. (z.B. um größere / kleinere Dinge besser sichtbar zu machen)

Wenn Gefäße mit non-uniformen Kontrastmittel sichtbar gemacht werden, ist das Rendering teils unzufriedenstellend. Lösung: Die TF zu „shiften entsprechend ihrer Gefäß-haftigkeit“.

## Modell-Freie Methoden

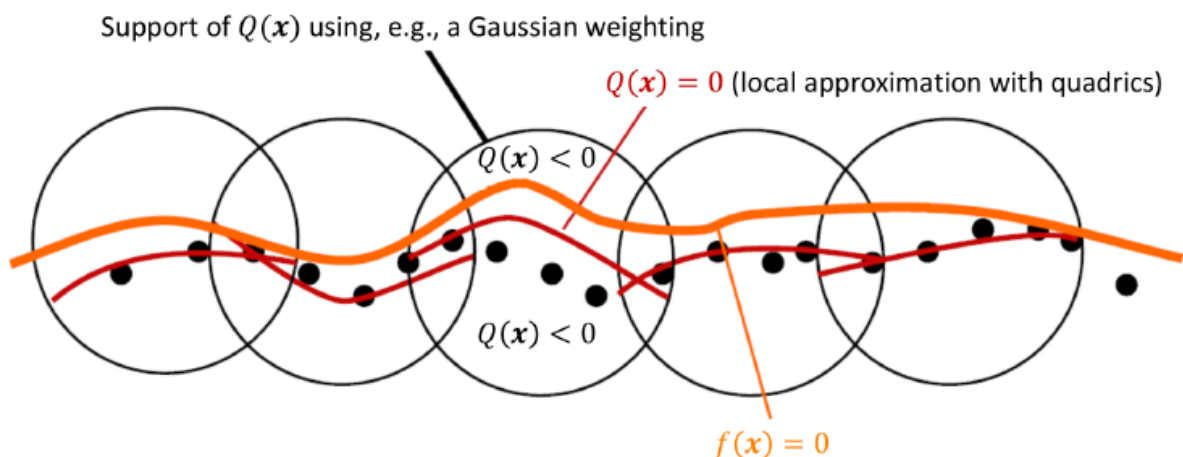
Motivation: Gefäßschnitt ist nicht perfekt kreisgeformt, wenn pathologisch verändert schon garnicht.  
→ Entfernung von Modell

MPUs

„Multi-Level Partition of Unity Implicits“. Vorbedingung für die Methode ist, dass die Gefäßdaten bereits durch **Bounding Voxels** begrenzt sind. (Preprocessing) Die begrenzten Punktwolken werden dann mit Oberflächen approximiert.

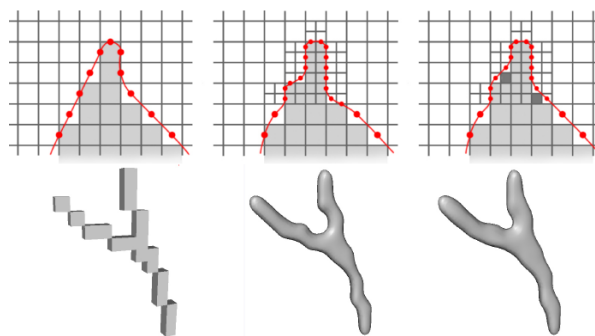
Ablauf:

1. Raum wird mit Oct-Tree aufgeteilt
2. Lokale Approximation durch Surfaces (Quadrics, kleinste Quadrate)
3. Globale Approximation durch Blending der lokalen Approximationen



Die Aufteilung beim Oct-Tree verläuft adaptiv „nach Normalverteilung“ (wut?)

**Problem:** Dünne Gefäß degenerieren. **Lösung:** Adaptives supersamplen.

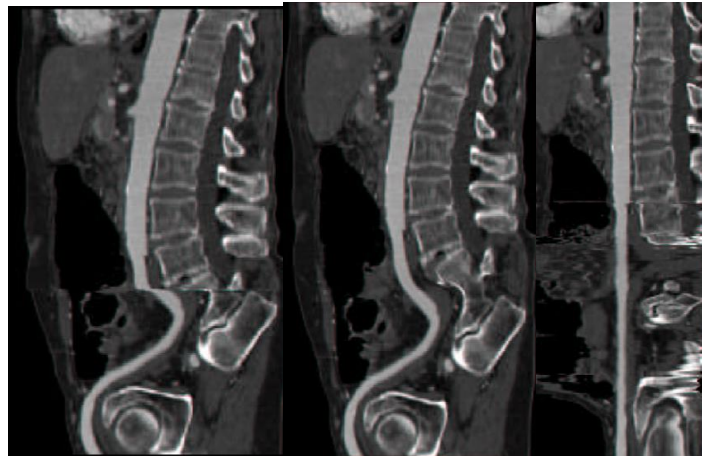


# Diagnostik

## Curved Planar Reconstruction

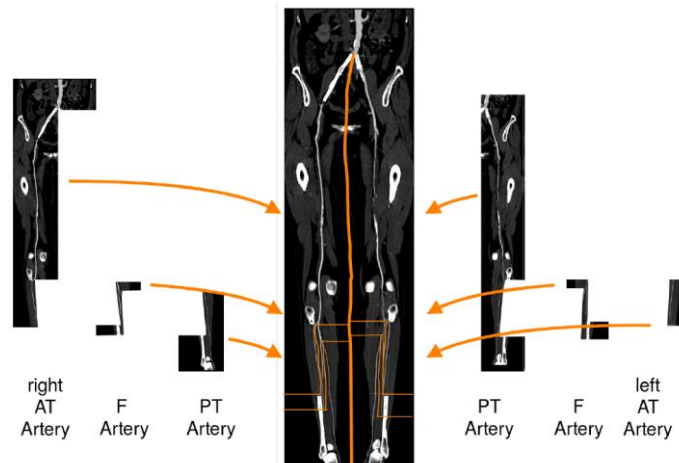
Man möchte das innere des Gefäßes (=lumen) untersuchen, also versucht man quasi eine Ebene durch das Gefäß zu legen, die dem Verlauf des Gefäßes folgt. → Image based Approach. Für die Darstellung dieser Ebene (die gekurvt ist) als flaches 2D Bild gibt es **drei Herangehensweisen**:

1. **Projiziert**: Am häufigsten, gute Räumliche Wahrnehmung
2. **Gestreckt**: Keine Okklusion
3. **Begradigt**: keine Okklusion, gerader Gefäßverlauf



## Multi-path CPR

Berechne CPR für jedes Gefäß-Segment, kombiniere alle in ein gemeinsames Bild. → Image based  
Hier wird nur projiziert oder gestreckt.

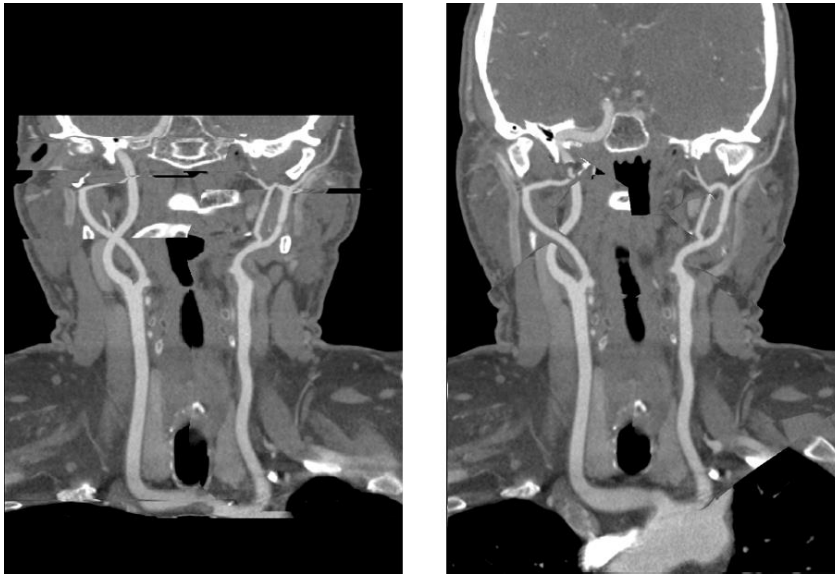


## Curved Surface Reformation (CSR)

Gehört quasi zur Gruppe der „projizierten CPRs“. Gefäße werden als wichtiger angesehen, wenn sie näher zum Betrachter sind und weiter Weg von einer Okklusion. (also freier liegen) Penalizing per Kostenfunktion.

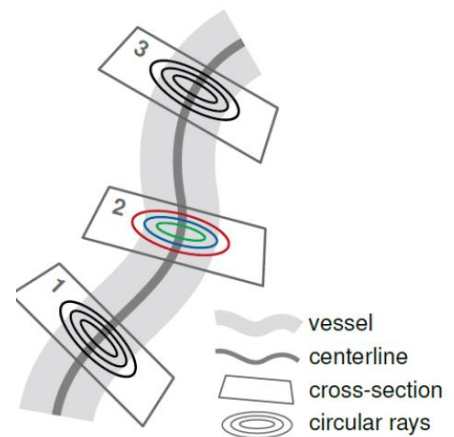
1. Erlaubt **freie Navigation** im Gegensatz zu CPR.
2. **Volle Sichtbarkeit multiple Gefäße**
3. **3D**, nicht Image based

## mpCPR vs. CSR



### Tools und Weiteres

- **VesselGlyph** kombiniert DVR (gibt Kontext) und CPR (gibt Fokus), zeigt Gefäße Okklusionfrei innerhalb anatomischen Kontextes
- **Untangled CPR**: CPR von kompletten Gefäßsystem ohne Okklusionen in einem Bild. (projiziert Gefäßsegmente in Image-Space, gestreckte CPRs werden geordnet per Minimierung von Schnitten der Konvexnhpflen
- **Curvicircular Feature Aggregation (CFA)**: =Non-linear Raycasting. Aggregiert bereits rotierte CPRs in einem Bild. Erlaubt auch Gruppierung und Anzeige von Gefäßen nach ihrer anatomischen Lokation (siehe Bild)



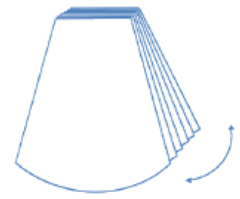
## Kapitel 7: Visualisierung von fetalem Ultraschall

Ultraschall generell ist Schall  $>20$  KHz, in der Medizin meist **2-18 MHz**. Funktioniert simpel über **Echolation**. Der Wandler schickt Ultraschall ins Gewebe, welches den Schall zurückechoet. Das kann der Wandler empfangen und Verarbeiten. Konversion läuft über **piezoelektrische Element** (Strom $\leftrightarrow$ Schall)

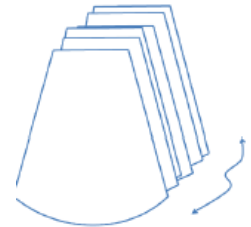
Der Schall breitet sich vom Wandler in konstanter Geschwindigkeit aus, wird Gewebe erreicht so wird der **Schall gestreut und gedämpft**. Bei **Übergängen** zwischen **verschiedenen Gewebsarten** (=unterschiedliche akustische Impedanz) wird der **Schall als Echo reflektiert**. (=Echogenität)

Bei Ultraschall gibt's 2 Modi:

- **A-Modus:** Amplitude des Signals bei Ausbreitung. Weißt auf unterschiedliche Gewebearten hin. (Luft, Wasser, Knochen, ...)
- **B-Modus:** Brightness, Helligkeit. Hierbei wird ein 2D Grauwert-Bild generiert. Jeder Pixel entspricht hierbei der Amplitude an der entsprechenden Position.



**3D-Bilder** werden beim Ultraschall aus **mehreren B-Bildern** zurückgerechnet. Der Raum entsteht durch manuelles bzw. maschinelles Bewegen des Wandlers. Alternativ gibt's auch Wandler bei denen die piezoelektrischen Elemente einen 2D-Grid bilden.



Indem **mehrere Bilder** aufgenommen werden, kann **auch** (nicht perpendikulare) **Bewegung** aufgezeichnet werden. Die **Echo Positionen werden automatisch gematched** und Bewegung hinzu und weg vom Wandler kann somit detektiert und Blau/Rot kodiert werden. (Geschwindigkeit und Richtung, wird Eingesetzt um Blutfluss darzustellen)

Mit mikroskopischen „Luftbläschen“ können Bilder verbessert werden.

#### Pros:

1. Kann viele **verschiedene Gewebsarten** aufzeichnen
2. **4D-Aufnahmen** möglich (Raum+Zeit)
3. **Billig**, leicht einzusetzen (MRI/CT viel Aufwändiger)
4. **Nicht invasiv** (sonst wären Fetalbilder auch nicht möglich)

#### Cons:

1. Es gibt viele **Artefakte**
  - a. **Speckle:** Random, granulär. Entsteht durch Welleninterferenz (Streuung)
  - b. **Dämpfungen:** z.B. Knochen werfen Schatten (starke Dämpfung)
  - c. **Reverberation** (Echoüberlagerungen)
  - d. Artefakte wegen gleichzeitiger **Bewegung von Wandler UND Gewebe**
2. Braucht einen **fähigen Operator** der die Aufnahmen erstellt
3. Benutzt **keine regulären Datagrids**, sondern konische bzw. sektorale
4. **Auflösung ist positionsabhängig** in B-Modus

#### Simple Visualisierungen

1. **Konventionelle orthogonale Schnitte**
2. **Curvi-planare reformierte Slices** (Länge, Größe und Position)
3. **MIP** (besserer anatomischer Kontext)
4. **DVR** (overall Shape, Gut um es den Eltern zu zeigen)

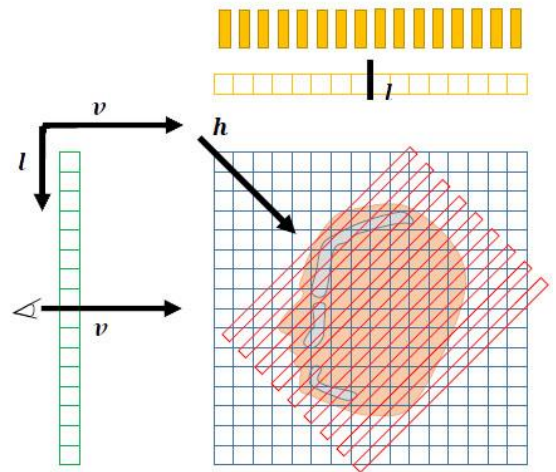
## Fortgeschrittene Ultraschallvisualisierung

Algorithmus basierend auf „half-angle slicing“: Ästhetisch ansprechend, stellt sehr gut unterschiedliche Gewebsarten dar und kann dank realistischer Beleuchtung gut Merkmale und des Fötus betonen.

Zur Erklärung des **Algorithmus** im Detail: Siehe Folien.



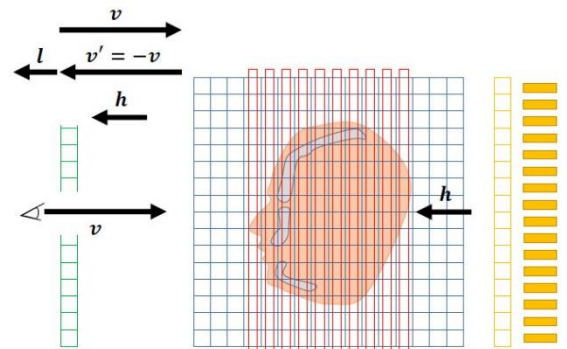
1. Wenn der Winkel zwischen  $l$  und  $v \leq 90^\circ$  ist, dann wird der **Halbvektor**  $h$  definiert als  $h = (v+l)/2$
2. Das Volumen wird nun **perpendikular zu  $h$**  gesliced.
3. **Zuerst** werden die **Light-Slices** generiert (gelbes Array, oben)



- a. **Alle Felder** werden **Weiß** initialisiert
  - b. Die eintreffende **Illumination  $L$**  wird mit der **Transportfarbe des Gewebes  $C_{tr}$**  multipliziert um die Lichtkontribution für jeden Voxel  $L'$  zu berechnen
  - c. Die **eintreffende Illumination  $L$**  wird von der Lightslice gemappt und das **Ergebnis der Multiplikation** in  $L'$  in die Lightslice **gespeichert**.
4. **Nun** werden die **Bild-Slices generiert** (grünes Array, links)
    - a. **Alle Felder** werden **Transparent** initialisiert
    - b. Die **Illumination** für einen geshadeten Voxel  $L$  wird von der **korrespondierenden Lightslice** genommen
    - c. Die **geshadete Farbe  $C_{res}$**  wird **berechnet** durch Multiplikation der Illumination  $L$  mit der diffusen Farbe des Voxels  $C_{diff}$
    - d. **Front-to-back compositing** wird eingesetzt um die geshadete Farbe  $C_{res}$  eines Bild-slice zu bekommen

Was wenn der Winkel zwischen  $l$  und  $v > 90^\circ$  ist?

- $v: v' = -v$  (Reverse Viewvektor)
- Halbvektor  $h$  demfalls per  $h = (v' + l)/2$
- Compositing verläuft Back-to-Front



## Compositing Modes

Das Compositing legt den Bild-Slices inhalt fest. (Bzw. fügt „Content“ hinzu)

- **FTB:** Layer der hinter dem Bild liegt wird hinzugefügt
- **BTF:** Layer der vor dem Bild liegt wird eingeblendet

Beide Modi starten mit **volltransparentem Bild**.

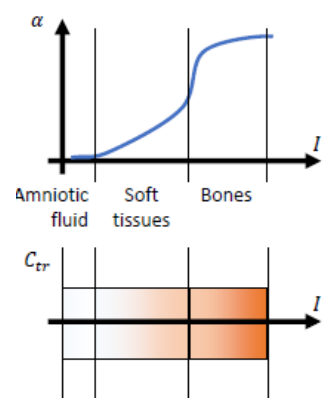
When we combine the layer with color  $X_{src} = (C_{src}, \alpha_{src})$  with the image color  $X_{dest} = (C_{dest}, \alpha_{dest})$

- ▶ Front-to-back compositing uses the following formula:

$$X_{dest} \leftarrow X_{dest} + (1 - \alpha_{dest}) * X_{src}$$

- ▶ Back-to-front compositing operates with the formula  $X_{dest} \leftarrow (1 - \alpha_{src}) * X_{dest} + X_{src}$

Die **Transportfarbe** ist ein Ausdruck um die **optischen Eigenschaften einer bestimmten Gewebeart zu beschreiben**. Es entspricht einer **Farbe nachdem weißes Licht durch ein Sample des Gewebes** ging. Rechts ist die Funktion kontinuierlich, da sich Knochen nicht klar von weichem Gewebe trennen lässt. Amniotische Flüssigkeit ist das einzige optisch volltransparente Medium.



Half-Angle Slicing hat einen **hohen diagnostischen Wert**: Malformationen bereits im ersten Trimester lassen sich damit identifizieren. Bestimmte Charakteristika z.B. für Down-Syndrom lassen sich gut mit der Visualisierung erkennen.

Ein letztes Problem ist noch, dass bei fetalen Ultraschall das **Gewebe der Mutter stark zu Okklusion** (und damit Artefakten/Noise) **führt**. Man muss also irgendwie das Gewebe der Mutter von dem des Fötus unterscheiden. → Klassifikation

1. Wandler schickt Schall in den Bauch, versucht **Knochen** (Gesichtsknochen des Fötus) zu **detektieren**.
2. Sind die Knochen aufgefunden wird versucht ein **leerer Raum zwischen Mutter und dem Fötus** zu finden.
3. Ist dieser Raum gefunden werden darin **mittige Punkte** gelegt die letztendlich kombiniert und daraus eine **Clipping-Plane** erstellt wird.

## Kapitel 8: Molekular Visualisierung

Durch Molekular Visualisierungen lassen sich chemische Zusammensetzungen verstehen und studieren, dies ist ein wichtiger Teil zur Entwicklung neuer Medikamente.

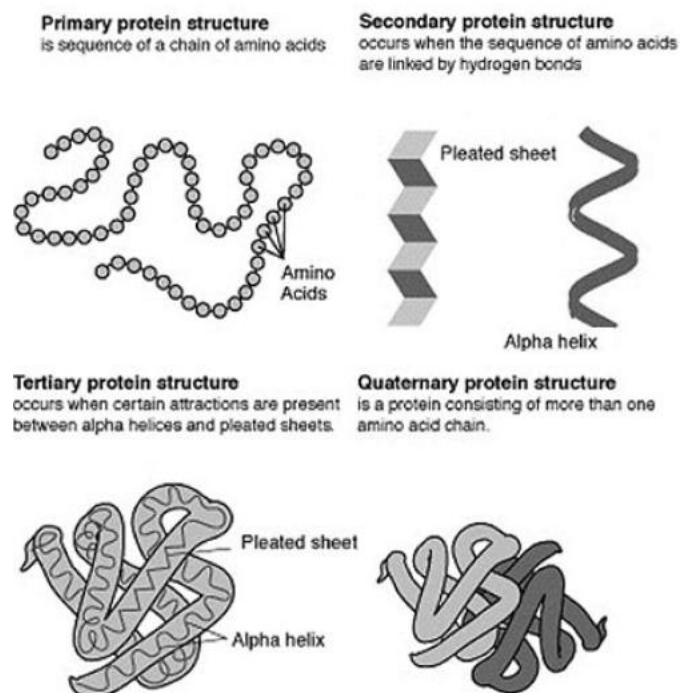
### Kleine Wiederholung

Atome haben einen **Nukleus (Protonen/Neutronen)** und **Elektronen im Orbit**. Bindungen können kovalent, ionisch, hydrogen oder disulfid sein. Die DNA speichert **genetischen Code** als zur **Generation von Proteinen**. DNA selbst eine Kette von Nukleotiden, 3 davon kodieren eine Aminosäure und die sind wiederum die Basis für Proteine. Proteine haben Primär-, Sekundär-, Tertiär und Quarternärstruktur.

Die Aminosäuren verbinden sich per **Peptidverbindungen** zu Proteinen. Der Backbone-Teil ist immer in N-C-C. Der Restteil wird wirklich als „Rest“ bezeichnet. Die Reste haben unterschiedliche Grade an Freiheit. Es gibt 20 „Standardamino-säuren“.

### Simulationen:

- **Molekular Dynamik**: Newtonsche Physik wird simuliert. Keine neuen Verbindungen werden erstellt.
- **Normal Modus Analyse**: Eine Atomposition pro Rest kalkuliert.



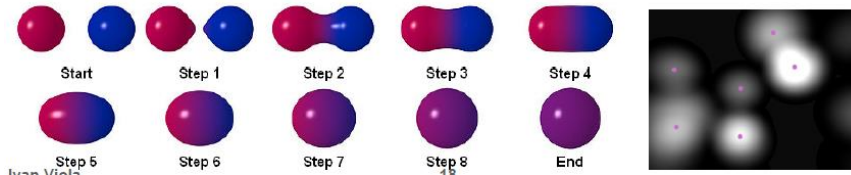


## Darstellungsformen

### Gaussche Oberflächen

Definition eines gausschen „3D-Blobs“

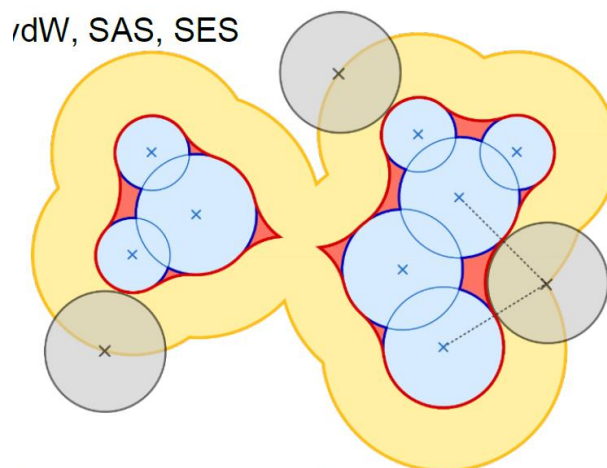
$$G(x, y, z) = \sum_{i=1}^n g_i(x, y, z) \geq t$$



Die Atome werden diskretisiert in einem volumetrischen Grid. Iso-Surface wird per Ray-Casting extrahiert. Binary-search für Schnittstellen-Test:  $G(x, y, z) = \sum_{i=1}^n g_i(x, y, z) \geq t$

### Solvent Accessible Surface (SAS) / Solvent Excluded Surface

Darstellungsformen für Oberfläche von Molekülen. Zur Erstellung wird eine Probe-Atom (Wasserstoff) entlang der Atome des Moleküls bewegt und daran die Oberfläche generiert.



Blau: Van der Waals, Gelb: SAS, Rot: SES

### SES: Darstellungsmöglichkeiten

1. **Grid-basiert:** SAS wird in Volumen gerendert mit „1“, bei jeden Oberflächen-Punkt setze Sphäre und überschreibe mit 0. (Ich denk mal man Zeichnet zuerst das Gelbe und entfernts dann quasi wieder sodass nur das Rote bleibt)
2. **Analytisch:**
  - a. **Reduzierte Oberflächen**
  - b. **Konturbildung**
  - c. **Real-time Ray-casting von „Implicits“**

### SES: Reduced Surface

1. Input: Ein **Set M** von **n Sphären** und eine **Probesphäre  $r_p$**
2. Anfangs wird ein initial-Atom ausgewählt und weitere aus dessen Nachbarschaft
3. Es wird ein **initiales RS Face** konstruiert (flaches Dreeick, welches sich aus den Zentren der Sphären c1,c2 und c3 ergibt)

4. Von hier aus rollt die **Probe Sphäre über eine RS Edge** (Kante des RS Face, z.B. die direkte Linie zwischen  $c_1$  und  $c_2$ ) bis sie auf eine **weitere Sphäre trifft**.
5. Daraus resultiert ein **neues RS Face**, was wiederum **neue RS Edges** für den rekursiven Algorithmus bedeutet
6. Dies läuft so fort, **bis keine neuen Edges mehr auffindbar** sind und alle RS Face identifiziert wurden
7. Für jedes **RS Face** kann ein **sphärisches Dreieck** erzeugt werden durch **Schnitt der** Probensphäre  $A_s$  mit der **Ebene** von **ihrem Zentrum  $C_s$**  und den **Zentren der angrenzenden Atome  $C_i, C_j, C_k$**  aufgespannt wird.
8. Für alle sphärischen Dreiecke müssen nun **toruidiale (torusförmige) Patches** gefunden werden
9. **Jede RS Edge** definiert einen Tours. Die RS Edge ist die **Rotationsachse**. Wenn die Probensphäre von einer Position  $c$  zur nächsten  $c'$  rollt beschreibt sie ein „Teilstück“ eines Kreises bzw. einer Kugel. Verbindet man diese Stücke mit den Eckpunkten der angrenzenden sphärischen Dreiecke, entsteht der gesuchte Torus.

#### SES: Konturbildung

Rührt daher, dass SES Patches voll beschrieben werden von den Schnittpunkten der Sphären die SAS definieren.

1. Für jedes Atom  $a_i$  wird eine Kontur gebaut durch den SAS-Schnitt mit seinen Nachbarn
2. Die **Kontur wird inkrementell gebaut**, wenn sich **Konturen schneiden**, denn werden **Kreise(-konturen) in die sphärischen Teilstücke geschnitten**
3. Die **Kreuzungen** repräsentieren die **Zentren** der sphärischen Dreiecke und die **Konturen** repräsentieren den **Toruspatch**

Die letzte Methode ist ein Formel-Blabla, vergiss es.