Volume Visualization

Part 1 (out of 3)

Overview: Volume Visualization

- Introduction to volume visualization
  - On volume data
  - Surface vs. volume rendering
  - Overview: Techniques
- Simple methods
  - Slicing, cuberille
- Direct volume visualization
  - Introduction, types of combinations
  - Transfer functions

Volume Visualization

- Introduction:
  - VolVis = visualization of volume data
    - Mapping 3D→2D
    - Projection (e.g., MIP), slicing, vol. rendering, …
  - Volume data =
    - 3D×1D data
    - Scalar data, 3D data space, space filling
  - User goals:
    - Gain insight in 3D data
    - Structures of special interest + context

Volume Data

- Where do the data come from?
  - Medical Application
    - Computed Tomographie (CT)
    - Magnetic Resonance Imaging (MR)
  - Materials testing
    - Industrial-CT
  - Simulation
    - Finite element methods (FEM)
    - Computational fluid dynamics (CFD)
  - etc.

3D Data Space

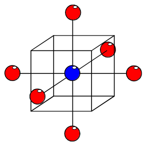
- How are volume data organized?
  - Cartesian resp. regular grid:
    - CT/MR: often dx=dy<dz, e.g. 135 slices (z) á 512^2 values (as x & y pixels in a slice)
    - **Data enhancement**: iso-stack-calculation = Interpolation of additional slices, so that dx=dy=dz $\Rightarrow$ 512^3 Voxel
    - Data: **Cells** (cuboid), Corner: **Voxel**
  - Curvi-linear grid resp. unstructured:
    - Data organized as tetrahedra or hexahedra
    - Often: conversion to tetrahedra

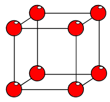
VolVis – Challenges

- Rendering projection, so much information and so few pixels!
- Large data sizes, e.g. $512\times512\times1024$ voxel á 16 bit = 512 Mbytes
- Speed, Interaction is very important, >10 fps!

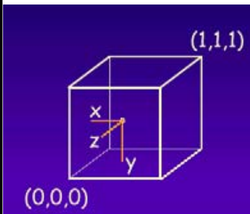
Voxels vs. Cells

- Two ways to interpret the data:
  - Data: set of voxel
    - **voxel** = abbreviation for volume element (cf. pixel = "picture elem.)
    - voxel = point sample in 3D
    - Not necessarily interpolated
  - Data: set of cells
    - cell = cube primitive (3D)
    - Corners: 8 voxel (see above)
    - Values in cell: interpolation used





Interpolation



$v = S(\mathrm{rnd}(x), \mathrm{rnd}(y), \mathrm{rnd}(z))$

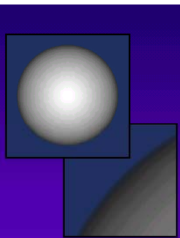
Nearest Neighbor

$$v = (1-x)(1-y)(1-z)S(0,0,0) + (x)(1-y)(1-z)S(1,0,0) + (1-x)(y)(1-z)S(0,1,0) + (x)(y)(1-z)S(1,1,0) + (1-x)(1-y)(z)S(0,0,1) + (x)(1-y)(z)S(1,0,1) + (1-x)(y)(z)S(0,1,1) + (x)(y)(z)S(1,1,1)$$

Trilinear

Interpolation – Results

Nearest Neighbor Interpolation



Trilinear Interpolation

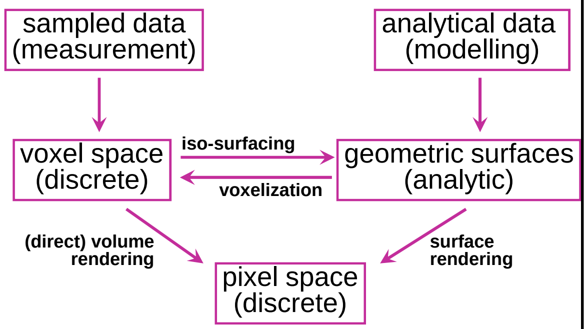
Gradients as Normal Vector Replacement

- Gradient $\nabla f = (\partial f/\partial x, \partial f/\partial y, \partial f/\partial z)$
- $\nabla f|_{x_0}$ normal vector to iso-surface $f(x_0)=f_0$
- Central difference in x-, y- & z-direction (in voxel):

$$\nabla f\,(x,y,z) \;=\; 1/2 \begin{pmatrix} f(x+1)-f(x-1) \\ f(y+1)-f(y-1) \\ f(z+1)-f(z-1) \end{pmatrix}$$

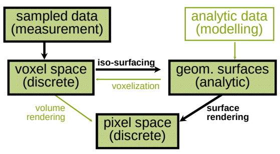
- Then tri-linear interpolation within a cell
- Alternatives:
  - Forward differencing: $\nabla f\,(x)=f(x+1)-f(x)$
  - Backwards differencing: $\nabla f\,(x)=f(x)-f(x-1)$
  - Intermediate differencing: $\nabla f\,(x+0.5)=f(x+1)-f(x)$

Concepts and Terms



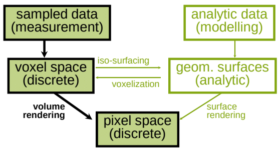
Concepts and Terms

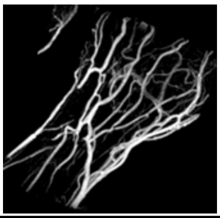
- Example 1:
  - CT measurement
  - Iso-stack-conversion
  - Iso-surface-calculation (marching cubes)
  - Surface rendering (OpenGL)



Concepts and Terms

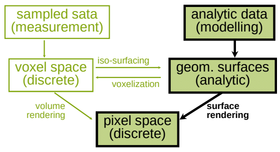
- Example 2:
  - MR measurement
  - Iso-stack-conversion
  - MIP (maximum intensity proj.)
  - Image: blood-vessels in hand

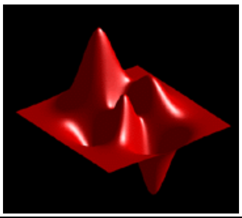




Concepts and Terms

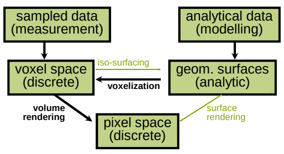
- Example 3:
  - potential function $\rho(x,y,z)$
  - Iso-surface $\rho(x,y,z)=\rho_0$
  - Surface: ray tracing

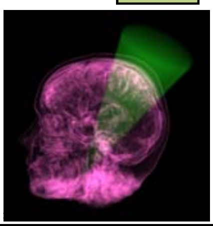




Concepts and Terms

- Example 4:
  - X-Ray Modelling
  - Surface-definition
  - Sampling (voxelization), combination
  - Direct volume rendering

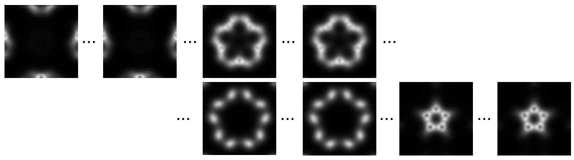




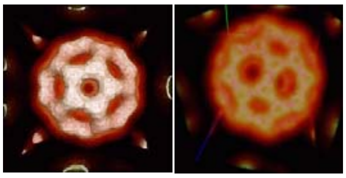
Surfaces vs. Volume Rendering

- Surface rendering:
  - **Indirect** volume visualization
  - Intermediate representation: iso-surface, “3D”
  - Pros: Shading→Shape!, HW-rendering
- Volume rendering:
  - **Direct** volume visualization
  - Usage of transfer functions
  - Pros: illustrate the interior, semi-transparency

Surfaces vs. Volume Rendering



hybrid rendering = surfaces +volumes



volume rendering

VolVis-Techniques – Overview

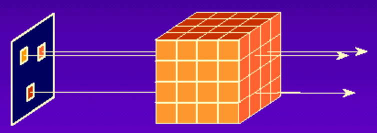
- Simple methods:
  - Slicing, MPR (multi-planar reconstruction)
- Direct volume visualization:
  - Ray casting
  - Shear-warp factorization
  - Splatting
  - 3D texture mapping
  - Fourier volume rendering
- Surface-fitting methods:
  - Marching cubes (marching tetrahedra)

Image-Order vs. Object-Order

- Image-order:
  - FOR every pixel DO: …
  - Cost, complexity ≈ image size
  - Example: ray casting (tracing viewing rays)
- Object-order:
  - FOR every object (voxel) DO: …
  - Cost, complexity ≈ object size (# of voxels)
  - Examples: splatting (“throwing snow balls")

Image-Order Approach

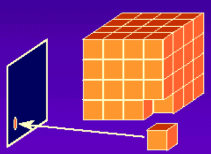
Image-Order Approach: Traverse the image pixel-by-pixel and sample the volume.



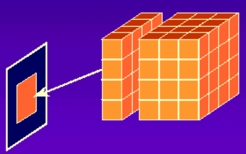
Ray Casting

Object-order approach

Object-Order Approach: Traverse the volume, and project to the image plane.



Splatting
cell-by-cell



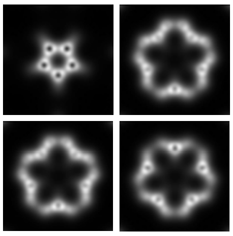
Texture Mapping
plane-by-plane

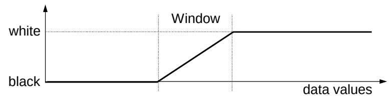
Simple Methods

Slicing, etc.

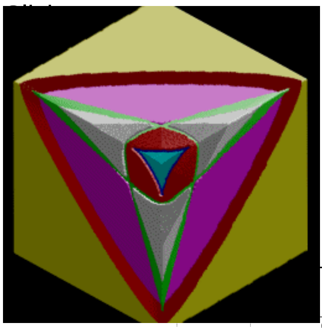
Slicing

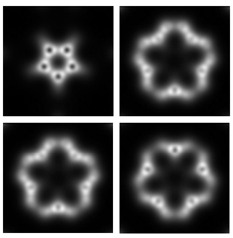
- Slicing:
  - Axes-parallel slices
  - regular grids: simple
  - without transfer function no color
  - Windowing:
    click! adjust contrast





Slicing

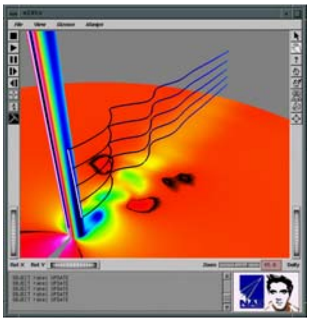




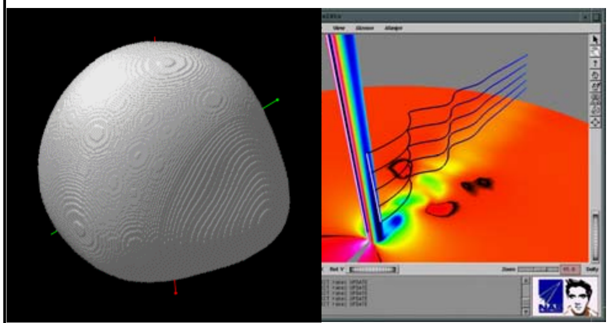
data values

Slicing

- Not so simple:
  - Slicing through general grid
  - Interpolation necessary
- Slicing:
  - well combinable with 3D-visualization
- Multi-planar reformation (MPR):
  - klick! arbitrary axes, 3D



Slicing



Direct Volume Visualization, Introduction

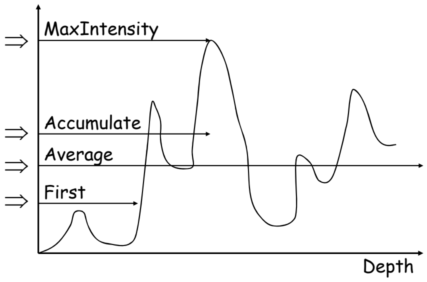
Classification – Transfer Functions

Direct Volume Visualization

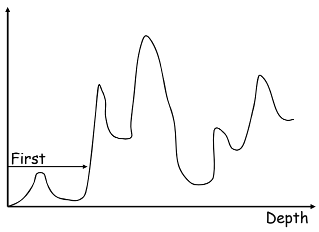
- Overview:
  - No intermediate representation
  - “real 3D”
  - Integration of so much information difficult
  - Object-order vs. image-order rendering
  - Various techniques (ray casting, splatting, shear-warp, texture mapping, Fourier volume rendering, etc.)
  - Various types of combinations (compositing, MIP, first-hit, average, etc.)

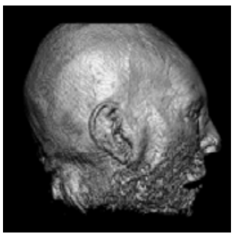
Types of Combinations

- Overview:
  - MIP ⇒
  - Compositing ⇒
  - X-Ray ⇒
  - First hit ⇒



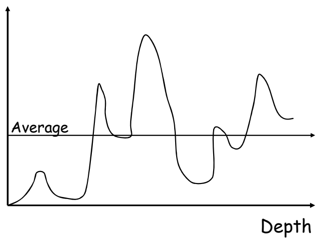
First Hit: Iso-Surface Extraction





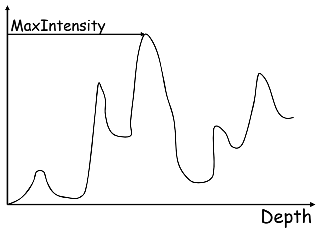
First: Extracts iso-surfaces (again!), done by Tuy&Tuy ’84

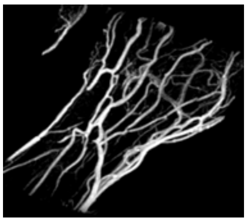
Average: as X-Ray Images



Average: Produces basically an X-ray picture

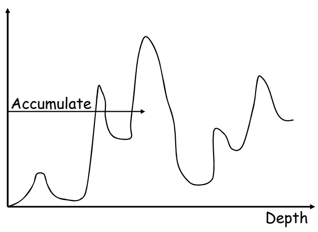
MIP: Maximum-Intensity Projection

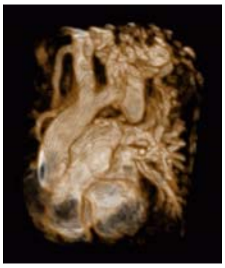




Max: Maximum Intensity Projection
used for Magnetic Resonance Angiograms,
for example

Compositing: Semi-Transparency

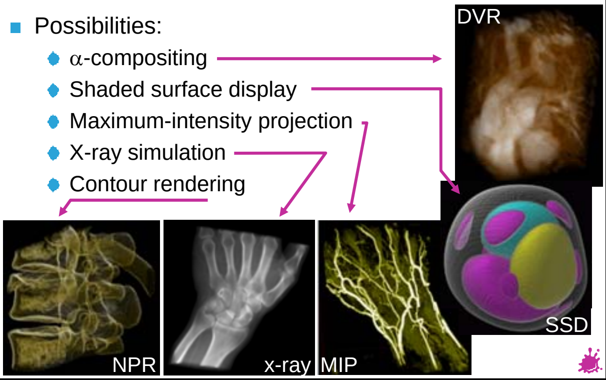




Accumulate: Make transparent layers visible!
Levoy ‘88

Types of Combination

- Possibilities:
  - α-compositing
  - Shaded surface display
  - Maximum-intensity projection
  - X-ray simulation
  - Contour rendering



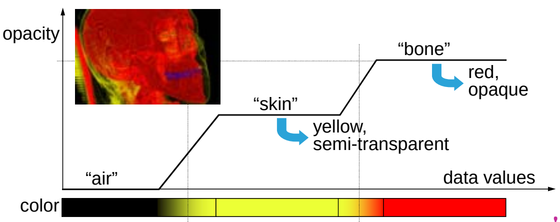
Classification

- Assignment data ⇒ semantics:
  - Assignment to objects, e.g., bone, skin, muscle, etc.
  - Usage of data values, gradient, curvature
  - Goal: segmentation
  - Often: semi-automatic resp. manual
  - Automatic approximation: transfer functions (TF)

Example

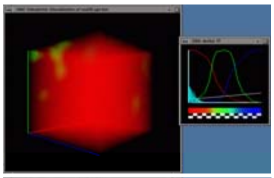
Transfer Functions (TF)

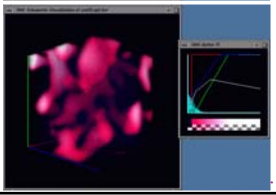
- Mapping data → "renderable quantities":
  - 1.) data → color
  - 2.) data → opacity (non-transparency)



Different Transfer Functions

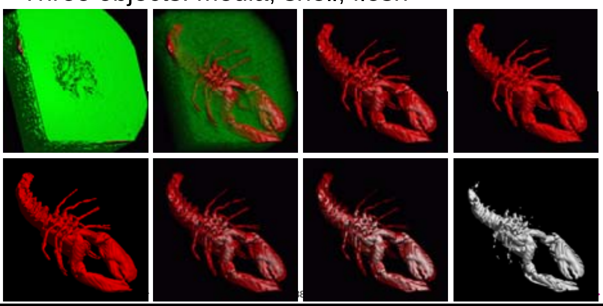
- Image results:
  - Strong dependence on transfer functions
  - Non-trivial specification
  - Limited segmentation possibilities





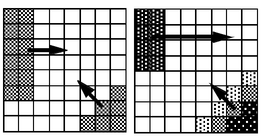
Lobster – Different Transfer Functions

- Three objects: media, shell, flesh



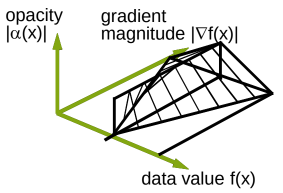
Inclusion of the Gradient

- Emphasis of changes:
  - Special interest often in transitional areas
  - Gradients: measure degree of change (like surface normal)
  - Larger gradient magnitude $\Rightarrow$ larger opacity



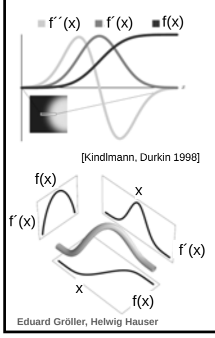
Gradient-Based Transfer Functions

- 2D-Transfer function:
  - Levoy ‘88
  - Specific opacity at certain threshold
  - but: close-by variation according gradient magnitude
  - highlights transitions (large gradients)
  - dampens homogeneous areas

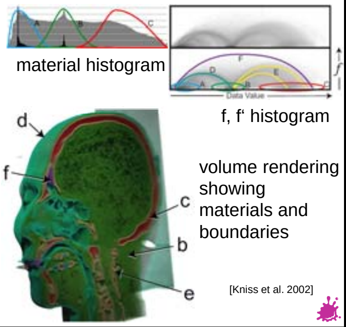


Multi-Dimensional Transfer Functions (1)

- f, f‘, f‘‘ histograms to depict material boundaries



[Kindlmann, Durkin 1998]



material histogram

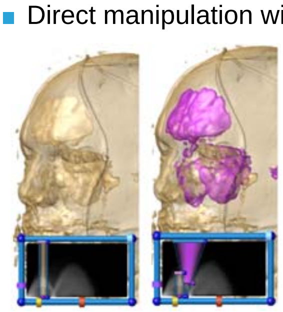
f, f‘ histogram

volume rendering showing materials and boundaries

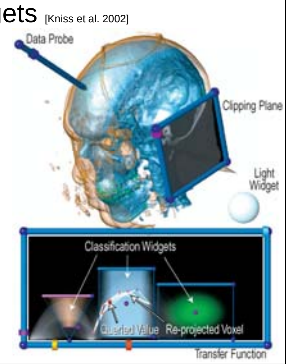
[Kniss et al. 2002]

Multi-Dimensional Transfer Functions (2)

- Direct manipulation widgets [Kniss et al. 2002]



1D vs. 2D transfer function



Acknowledgments

- For material for this lecture unit
  - Roberto Scopigno,
    Claudio Montani (CNR, Pisa)
  - Hans-Georg Pagendarm (DLR, Göttingen)
  - Michael Meißner (GRIS, Tübingen)
  - Torsten Möller
  - Gordon Kindlmann
  - Joe Kniss
  - etc.