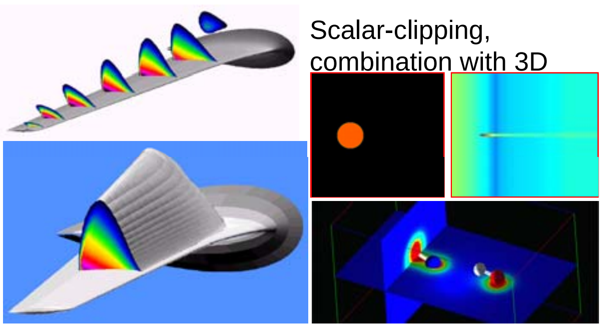
# Volume Visualization

Part 2 (out of 3)

## Review: Slices

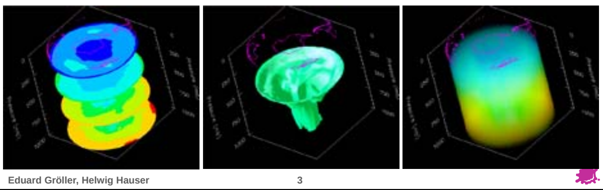
Scalar-clipping, combination with 3D





## Slices vs. Iso-Surfaces. vs. Volume Rendering

- Comparison ozon-data over Antarctica:
  - Slices: selective (z), 2D, color coding
  - Iso-surface: selective ($f_0$), covers 3D
  - Vol. rendering: transfer function dependent, "(too) sparse – (too) dense"

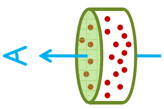


## Optical Models for Volume Rendering

Display of
Semi-Transparent Media

## Modelling of Natural Phenomena

- Various models (Examples):
  - Emission only (light particles)
  - Absorption only (dark fog)
  - Emission & absorption (clouds)
  - Single scattering, w/o shadows
  - Multiple scattering
- Two approaches:
  - Analytical model (via differentials)
  - Numerical approximation (via differences)





## Emission, Differential Model

- Continuous emission model:
  - Question: how much light (**I** like intensity) is added along an infinitely short ray segment in the volume
  - Differential $d\mathbf{I}/dt = g(t)$ …
    volume emits light (corresponding to thickness)
  - Glow factor $g(t)$
  - Integration results in: $\mathbf{I}(s) = \mathbf{I}_0 + \int_{t \in [0,s]} g(t)dt$
  - Overall emission contrib.: $G(0,s) = \int_{t \in [0,s]} g(t)dt$
  - Unrealistic, because no absorption

## Emission, Numerical Approximation

- Discrete emission model:
  - Question: how much light (C like color) is added within a small, but finite volume extent
  - $C_i$ … contribution of vol. extent i (thickness 1) $\Rightarrow$ adding emission of extent i results in $Out_i = In_i + C_i \Leftrightarrow Out_i = Out_{i-1} + C_i$
  - Accumulation:
    $Out_i = In_j + C_j + \ldots + C_{i-1} + C_i$
    $Out_i = In_j + \Sigma_{j \le k \le i} C_k$
  - Example:
    pixel value = background + $\Sigma_{k \in N} C(\mathbf{ray}(k))$



## Emission Only

- Differential model:
  - $\mathbf{I}(s) = \mathbf{I}_0 + \int_{t \in [0,s]} g(t)dt$
- Discrete approximation:
  - $Out_s = In_0 + \Sigma_{s \ge k \in N} C_k$
- Example:



## Absorption, Differential Model

- Continuous absorption model:
  - Question: how much light (in % of $\mathbf{I}_0$) remains after traversal of ray segment through the volume
  - Differential $d\mathbf{I}/dt = -\tau(t)\mathbf{I}(t)$ …
    light ($\mathbf{I}$) is partially absorbed ($\tau$)
  - Extinction coefficient $\tau(t)$, e.g., 30%
  - Integration results in: $\mathbf{I}(s) = \mathbf{I}_0 \cdot \exp(-\int_{t \in [0,s]} \tau(t)dt)$
  - Total transparency: $T(0,s) = \exp(-\int_{t \in [0,s]} \tau(t)dt)$
  - Total absorption: $\alpha(0,s) = 1 - T(0,s)$

## Absorption, Numerical Approximation

- Discrete approximation model:
  - Question: how much light (in % of $\mathbf{I}_0$) remains after traversal of small, but finite volume extent
  - $\alpha_i$ … opacity of volume extent i (per unit)
    $\Rightarrow$ result after traversal of extent i
    $Out_i = In_i \cdot (1-\alpha_i) \Leftrightarrow Out_i = Out_{i-1} \cdot (1-\alpha_i)$
  - Akkumulation: $Out_i = In_j \cdot (1-\alpha_j) \cdot \ldots \cdot (1-\alpha_i)$
    $Out_i = In_j \cdot \Pi_{j \le k \le i}(1-\alpha_k)$
  - Unit sampling: unit distance between $\alpha_i$ samples!!



## Absorption Only

- Differential model:
  - $\mathbf{I}(s) = \mathbf{I}_0 \cdot \exp(-\int_{t \in [0,s]} \tau(t)dt)$
- Discrete approximation:
  - $Out_s = In_0 \cdot \Pi_{s \ge k \in N}(1-\alpha_k)$
- Example:



## Emission and Absorption

- Continuous model (no scattering):
  - At each position is given:
    - Emission g(t)
    - Extinction coefficient $\tau(t)$
  - Differential $d\mathbf{I}/dt = g(t) - \tau(t)\mathbf{I}(t)$
  - Emission g(t) attenuated by T(t,s)
  - Only Emission: $\mathbf{I}_0 \quad + \int_{t \in [0,s]} g(t)\, dt$
  - With Absorption: $\mathbf{I}_0 \cdot T(0,s) + \int_{t \in [0,s]} g(t) \cdot T(t,s)\, dt$
  - Emission und Absorption:

$\mathbf{I}_0 \cdot \exp(-\int_{u \in [0,s]} \tau(u)du) + \int_{t \in [0,s]} g(t) \cdot \exp(-\int_{u \in [t,s]} \tau(u)du)\, dt$

## Numerical Approximation

- Discrete model (compositing):
  - For each volume extent i:
    - Contribution $C_i$
    - Opacity $\alpha_i$, transparency $1-\alpha_i$
  - $Out_i = In_i \cdot (1-\alpha_i) + C_i \cdot \alpha_i$ (Std.-compositing)
  - Convex combination from background and own contribution
  - $Out_s = In_0 \cdot \Pi_{s \geq k \in N} (1-\alpha_k) + \Sigma_{s \geq k \in N} C_k \cdot \alpha_k \cdot \Pi_{s \geq l > k} (1-\alpha_l)$
  - Opacity-weighted colors: $C_i \cdot \alpha_i$ instead of $C_i$



## Emission and Absorption

- Differential model:
  - $\mathbf{I}(s) = \mathbf{I}_0 \cdot T(0,s) + \int_{t \in [0,s]} g(t) \cdot T(t,s)\, dt$
  - $\mathbf{I}(s) = \mathbf{I}_0 \cdot \exp(-\int_{u \in [0,s]} \tau(u) du) + \int_{t \in [0,s]} g(t) \cdot \exp(-\int_{u \in [t,s]} \tau(u) du)\, dt$
- Discrete Approximation:
  - $Out_i = In_i \cdot (1-\alpha_i) + C_i \cdot \alpha_i$ (Compositing)
  - $Out_s = In_0 \cdot \Pi_{s \geq k \in N} (1-\alpha_k) + \Sigma_{s \geq k \in N} C_k \cdot \alpha_k \cdot \Pi_{s \geq l > k} (1-\alpha_l)$

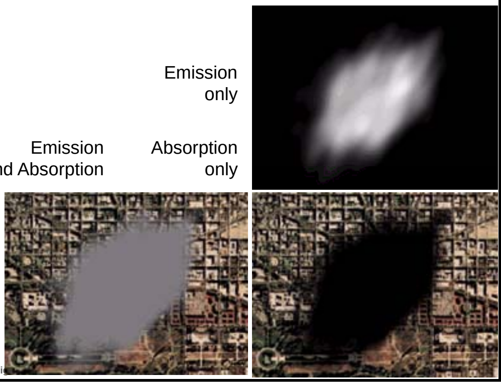


## Emission or/and Absorption

Emission only

Emission and Absorption

Absorption only



## Scattering

- **Scattering**: particles deviate light at a position
  - BRDF (bidirectional reflectance distribution function)
  - Single scattering
    - Too little light in the interior
  - Single scattering with shadows
  - Multiple Scattering
    - Radiosity techniques
    - Very realistic, very costly



## Literature

- Paper (more details):
  - Nelson Max: "**Optical Models for Direct Volume Rendering**" in *IEEE Transactions on Visualization and Computer Graphics*, Vol. 1, No. 2, June 1995



## Ray Casting / Compositing

Classical
Image-Order Methods

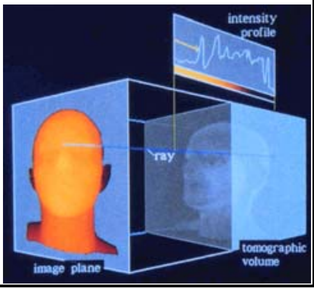
## Ray Tracing vs. Ray Casting

- Ray Tracing: method from image generation
- In volume rendering: only viewing rays
  ⇒ therefore Ray Casting
- Classical image-order method
- Ray Tracing: ray – object intersection
  Ray Casting: no objects, density values in 3D
- In theory: take all density values into account!
  In practice: traverse volume step by step
- Interpolation necessary for each step!



## Ray Traversal through Volume Data

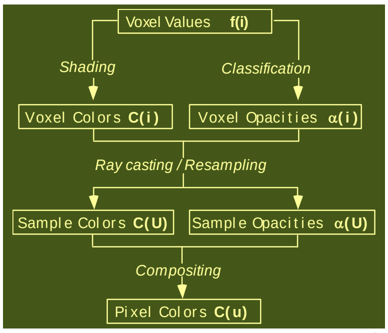
- Context:
  - Volume data: 1D value defined in 3D – $f(\mathbf{x}) \in R^1$, $\mathbf{x} \in R^3$
  - Ray defined as half-line: $\mathbf{r}(t) \in R^3$, $t \in R^1 > 0$
  - Values along Ray: $f(\mathbf{r}(t)) \in R^1$, $t \in R^1 > 0$ (intensity profile)





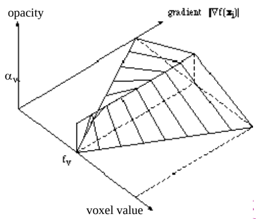

## Standard Ray Casting

- Levoy '88:
- 1. C(i), α(i) (from TF)
- 2. Ray casting, interpolation
- 3. Compositing



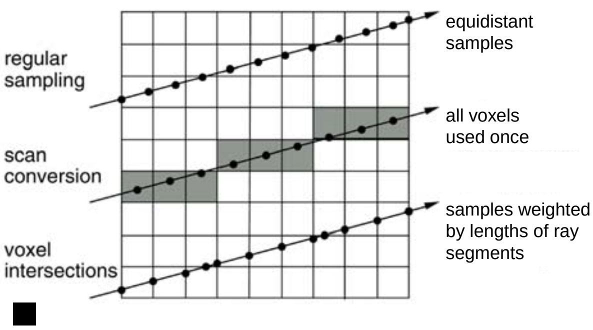
## 1. Shading, Classification

- 1. Step:
  - Shading, f(i)→C(i):
    - Apply transfer function
    - diffuse illumination (Phong), gradient ≈ normal
  - Classification, f(i)→α(i):
    - Levoy ’88, gradient enhanced
    - Emphasizes transitions






## 2. Ray Traversal – Three Approaches






## 2. Ray Traversal, Interpolation

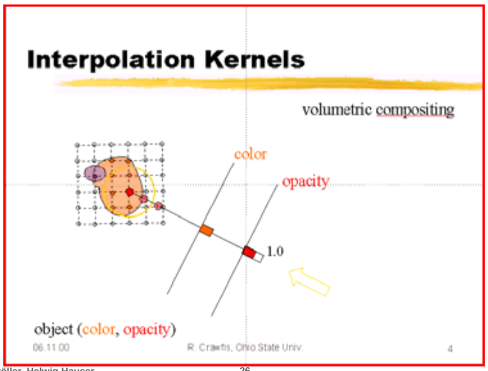
- Voxel-based vs. cell-based traversal
- Tri-linear (interpolation within a cell) vs. bi-linear (interpolation within a cell face)
- Tri-linear:
  - first 4* in z-direction (interpolated square),
  - then 2* in y-direction (interpolated line),
  - then 1* in x-direction (interpolated value)
- Unit sampling vs. variable sample distances – compositing different!!

## Compositing: F2B vs. B2F

- Back-to-Front (B2F):
  - $Out_i = In_i \cdot (1-\alpha_i) + C_i \cdot \alpha_i$, $In_{i+1} = Out_i$ …
  - Depending on local transparency $(1-\alpha_i) \Rightarrow$ convex combination of old $In_i$ & new $C_i$
  - Example:
    - Voxel i: $C_i$ = red, $\alpha_i$=30%; so far: $In_i$ = white
    - Result of compositing: 70% white + 30% red
- Front-to-Back (F2B):
  - $Col = Col + (1-\alpha_{akk}) \cdot C_i \cdot \alpha_I$ … accumulated color
  - $\alpha_{akk} = \alpha_{akk} + (1-\alpha_{akk}) \cdot \alpha_i$ … accumulated opacity



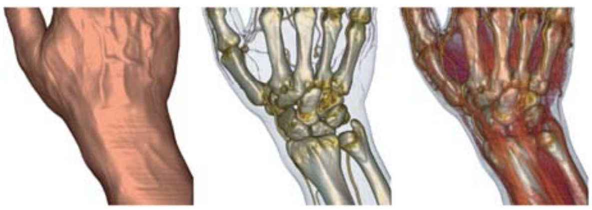
## Front-to-Back Compositing



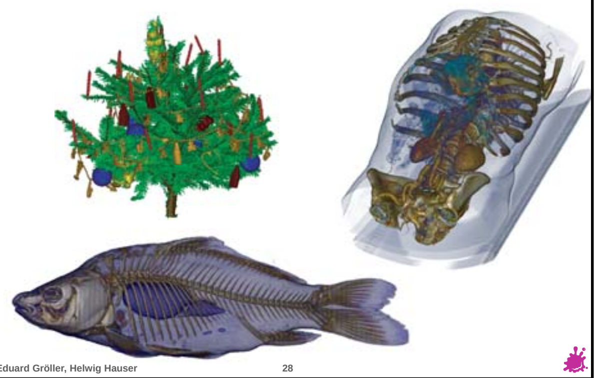



## Ray Casting – Examples

- CT scan of human hand (244x124x257, 16 bit)



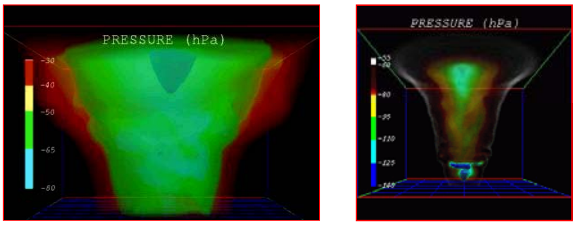
## Ray Casting – Examples





## Ray Casting – Further Examples

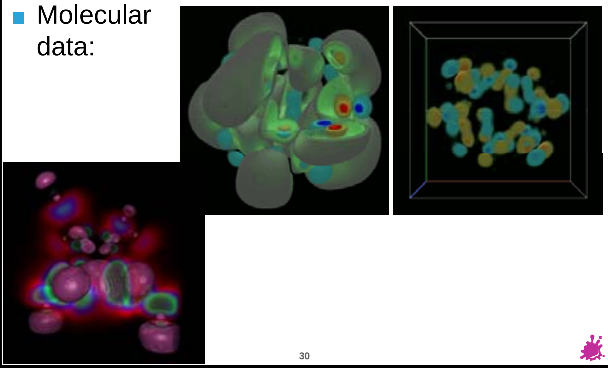
- Tornado Visualization:





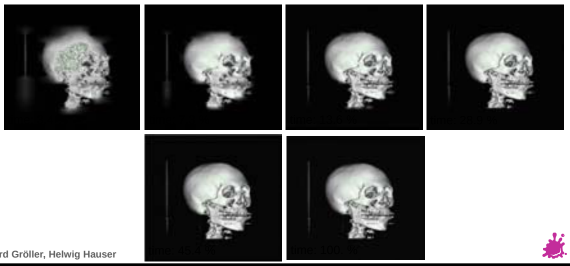

## Ray Casting – Further Examples

- Molecular data:



## Acceleration - Progressive Refinement

- First render every $2^n \times 2^n$-th pixel, then render the $2^{n-1} \times 2^{n-1}$-th pixel inbetween, aso. (until interruption or completion)





## Literature

- Paper (more details):
  - Marc Levoy: "**Display of Surfaces from Volume Data**" in *IEEE Computer Graphics & Applications*, Vol. 8, No. 3, June 1988



## Acknowledgement

- For material for this lecture unit:
  - Nelson Max (LLNL), Marc Levoy (Stanford)
  - Hans-Georg Pagendarm (DLR, Göttingen)
  - Lloyd Treinish (IBM)
  - Roberto Scopigno, Claudio Montani (CNR, Pisa)
  - Roger Crawfis (Ohio State Univ.)
  - Michael Meißner (GRIS, Tübingen)
  - Torsten Möller
  - etc.