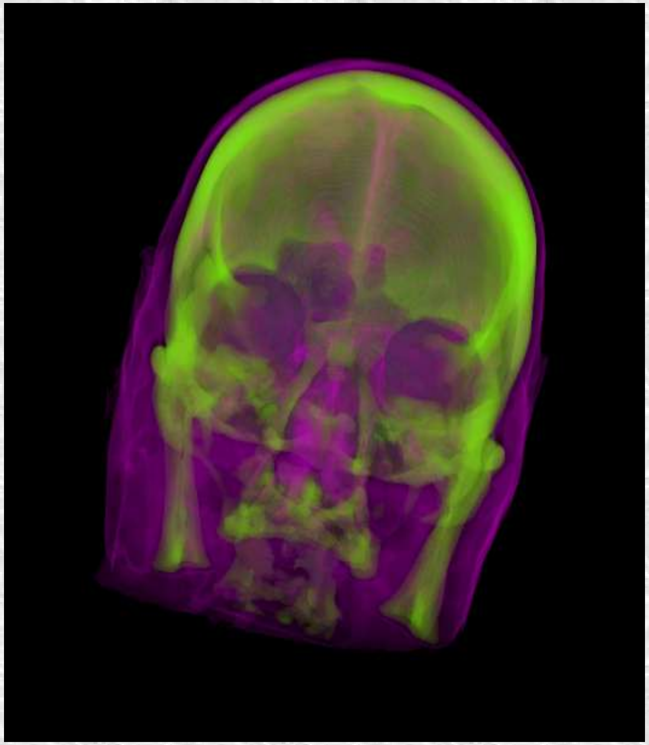
High-Quality Volume Graphics on Consumer PC Hardware



Dušan Migra

Supervisor : Ing. Miloš Šrámek, Dr.

Faculty Of Mathematics, Physics and Informatics

Data Representation

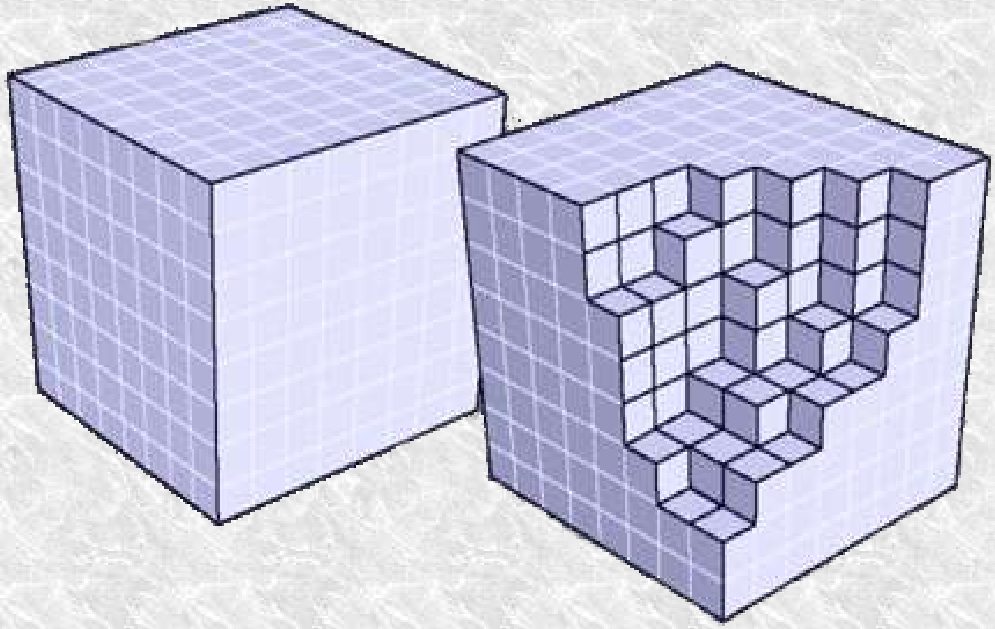
- **Volume Data** : stored as 3D texture or 2D texture stacks
- Continuous scalar field in 3D

$$s = f(x, y, z) \qquad x, y, z \in \mathbb{R}$$

- **Discrete volume**: voxels

$$s_i = v_i$$

- **Filtering** :
  - Tri-linear interpolation
  - Bi-linear interpolation
  - High-Order filtering



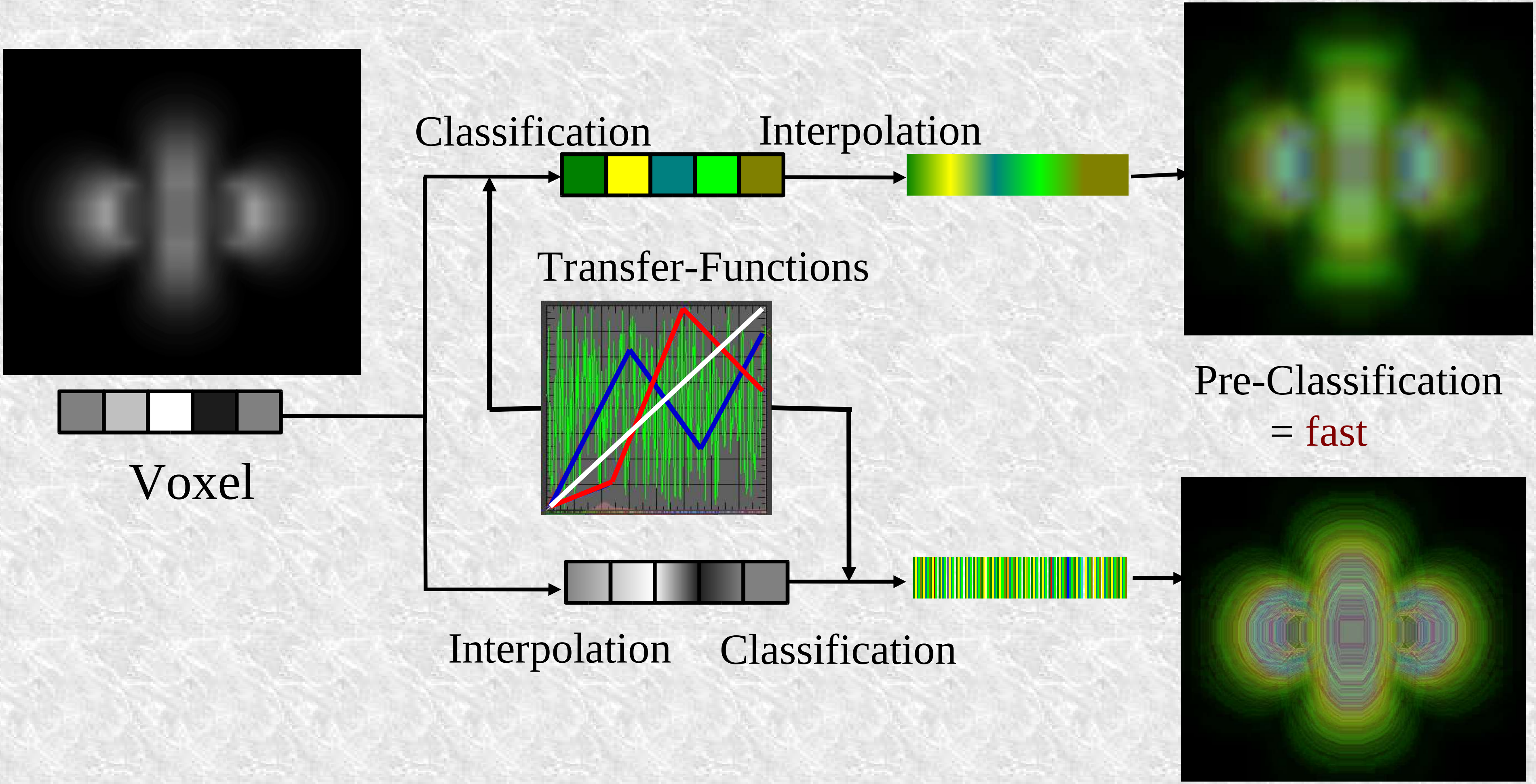
Direct Volume Rendering

- Direct methods
  - Image based (ray-casting)
  - Object based (cell-projection, shear-warp, texture-based)
- Volume rendering integral : $I = \int_0^D color(\boldsymbol{x}(\lambda))\mathrm{e}^{-\int_0^\lambda extinction(\boldsymbol{x}(\lambda'))d\lambda'} d\lambda$
- Map data values $s(\boldsymbol{x}(\lambda))$ to color $\tilde{c}(s(\boldsymbol{x}(\lambda)))$

  and extinction $\tau(s(\boldsymbol{x}(\lambda)))$ coefficients

$$I = \int_0^D \tilde{c}\left(s(\boldsymbol{x}(\lambda))\right)\mathrm{e}^{-\int_0^\lambda \tau\left(s(\boldsymbol{x}(\lambda'))\right)d\lambda'} d\lambda$$

Pre- and Post- Classification



Classification Interpolation

Transfer-Functions

Voxel

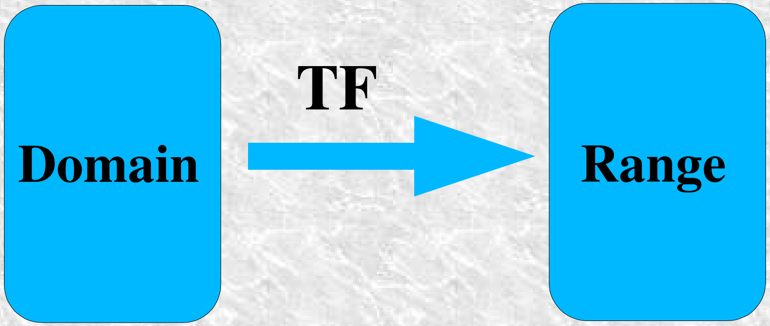
Interpolation Classification

Pre-Classification
= fast

Post-Classification
= quality

Transfer Functions

- Maps data values to colors $\tilde{c}(s)$ and $\tau(s)$ extinctions (opacities)



- Domain :
  - usually more than 8bit values
- Range
  - 8bit RGBA values

- Setting TFs if difficult, unintuitive, and slow

Numerical Integration

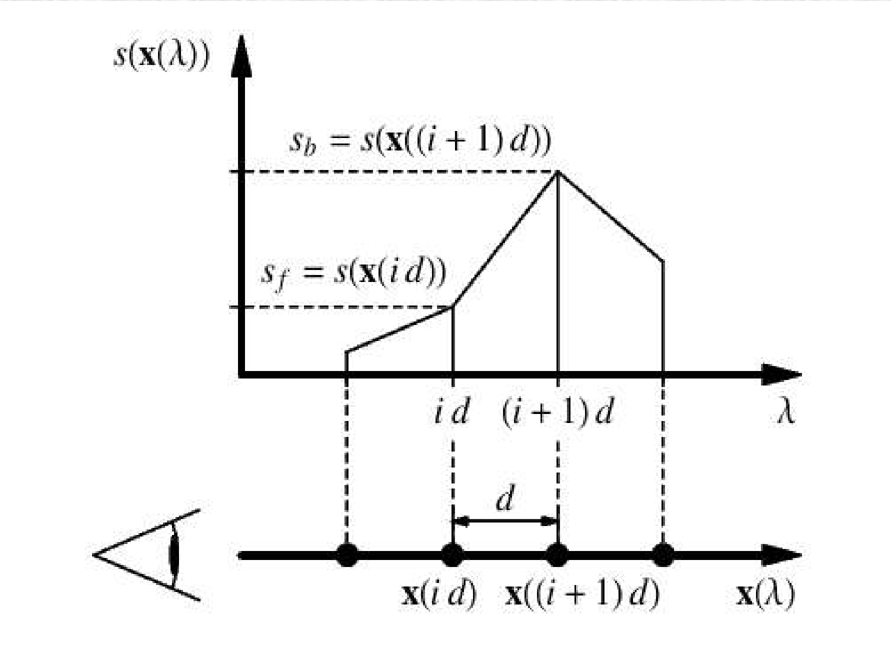
$$I=\int_0^D \tilde{c}\left(s(\boldsymbol{x}(\lambda))\right) e^{-\int_0^{\lambda}\tau\left(s(\boldsymbol{x}(\lambda'))\right)d\lambda'} d\lambda \approx \sum_{i=0}^{n}\tilde{C}_i\prod_{j=0}^{i-1}(1-\alpha_j)$$

$$e^{-\sum_{i=0}^{\lambda/d}\tau\left(s(\boldsymbol{x}(id))\right)d} = \prod_{i=0}^{\lambda/d} e^{-\tau\left(s(\boldsymbol{x}(id))\right)d} = \prod_{i=0}^{\lambda/d}(1-\alpha_i)$$

,where $\alpha_i \approx 1-e^{-\tau(s(\boldsymbol{x}(id)))}$

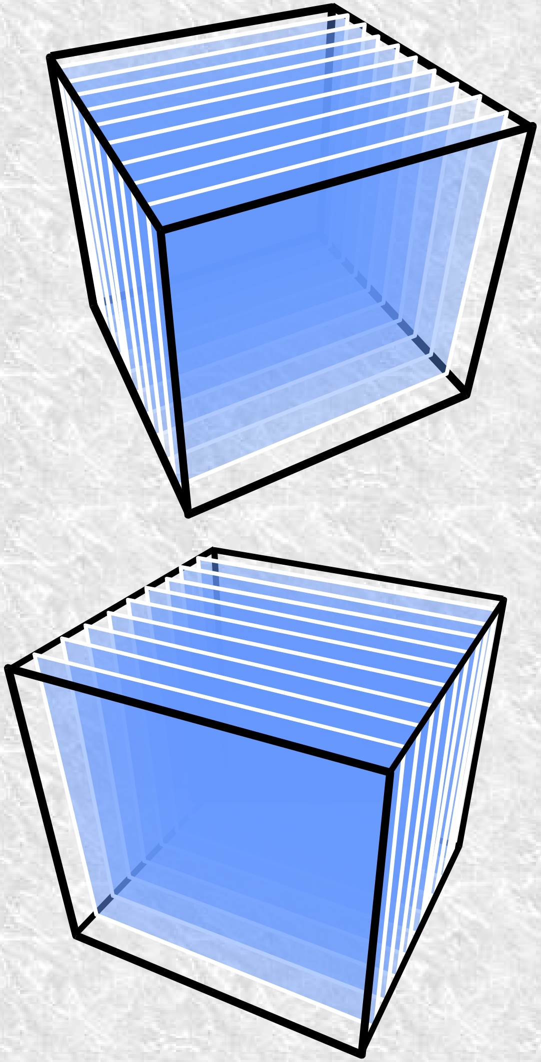
- Back-to-front algorithm

$$\tilde{C}'_i=\tilde{C}_i+(1-\alpha_i)\tilde{C}'_{i+1}$$

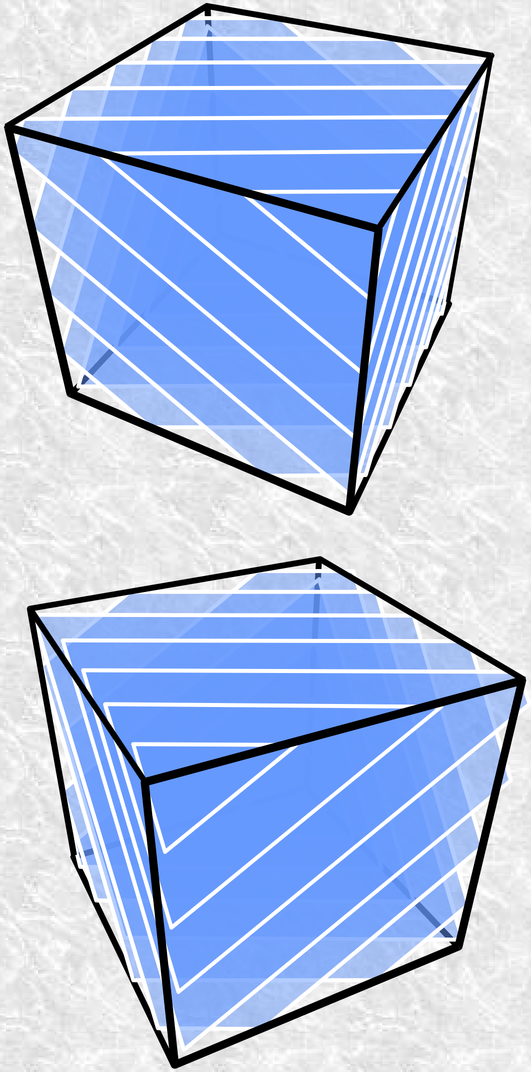


Texture-Based Volume Rendering

2D textured slices

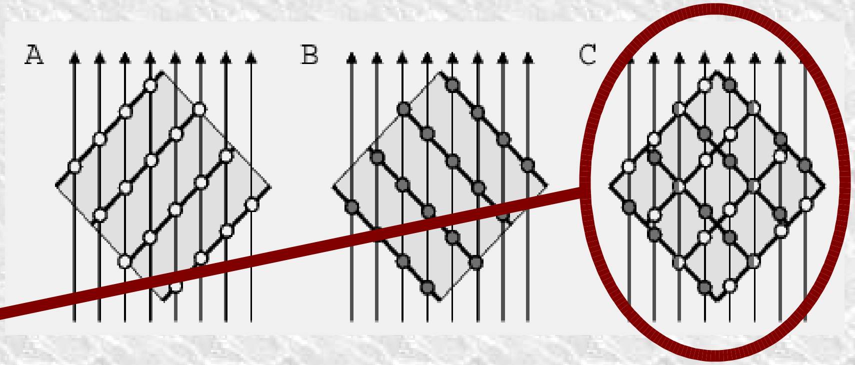


3D textured slices



Texture-Based Volume Rendering

- **2D Textures - object aligned slices (OAS)**
  - Bilinear interpolation in hardware
  - Three stacks needed
  - Inconsistent sampling rates
  - High performance
  - Switching effects



- **3D Textures - viewport aligned slices (VAS)**
  - Trilinear interpolation in hardware
  - Inconsistent sampling rates for perspective projection
  - High performance
  - No switching effects

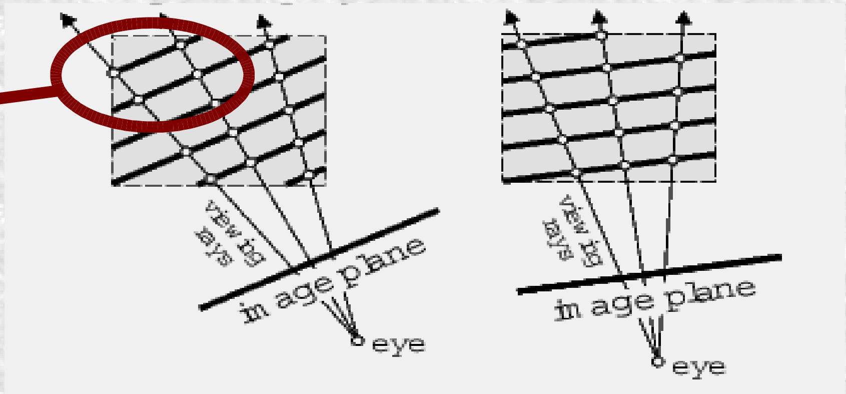
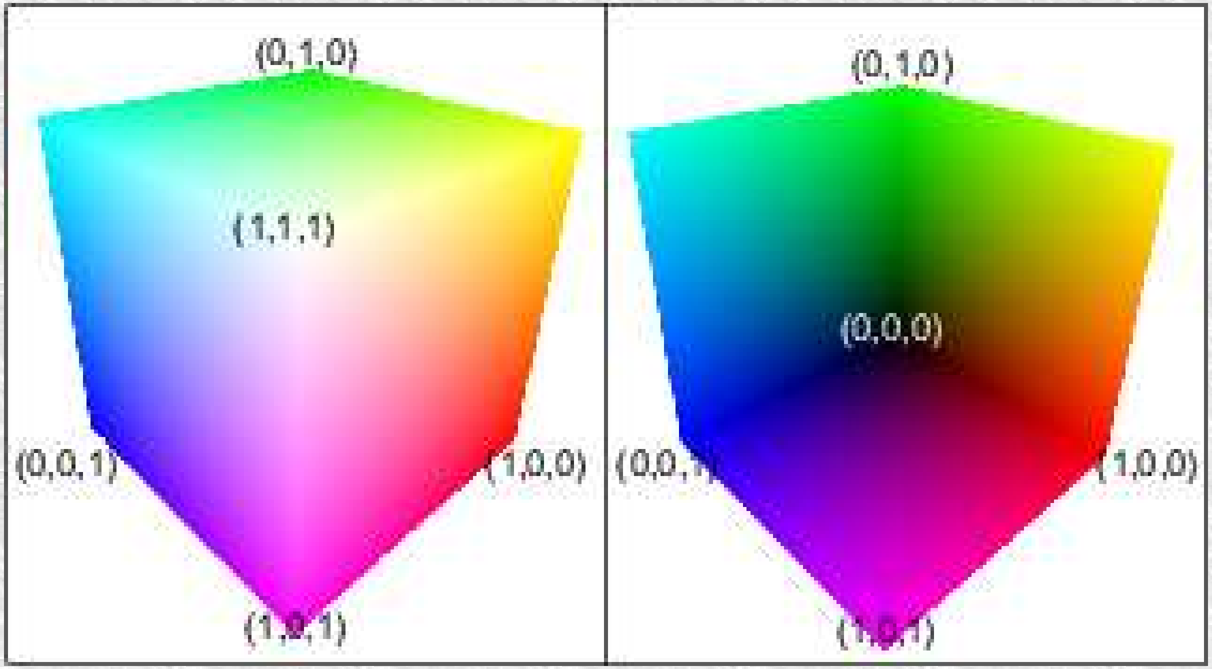


Image-Based Volume Rendering

- **Pass 1 :** Entry point determination
- **Pass 2 :** Ray direction determination
- **Main passes 3 to M :** Ray traversal
  - Rendering directed to a 2D texture



Pre-Integrated Volume Rendering

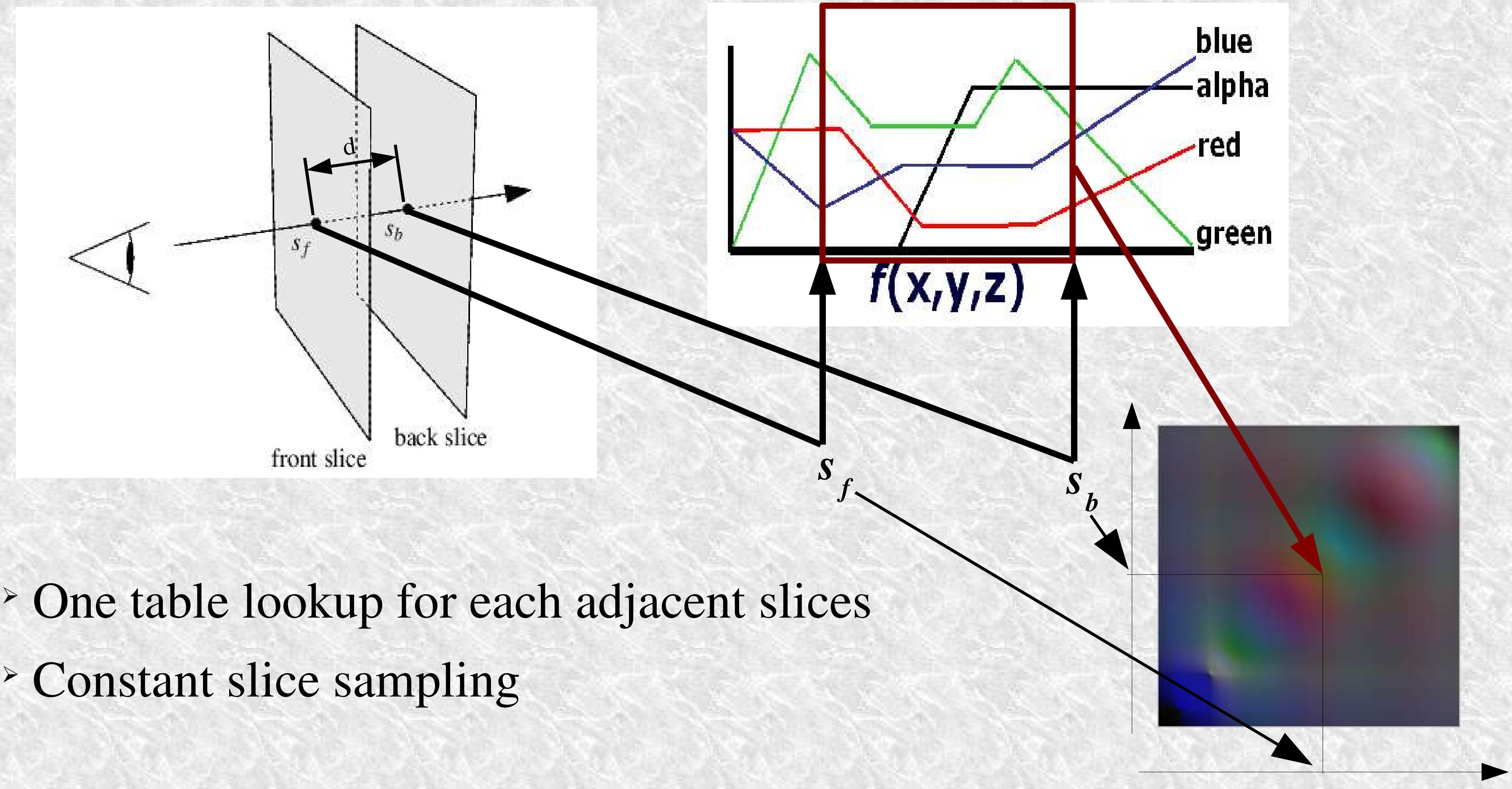
- **Idea** :

  Split the numerical integration into two integrations :

  - one for the continuous scalar field
  - one for the transfer functions $\tilde{c}(s)$ and $\tau(s)$

Pre-Integrated Classification

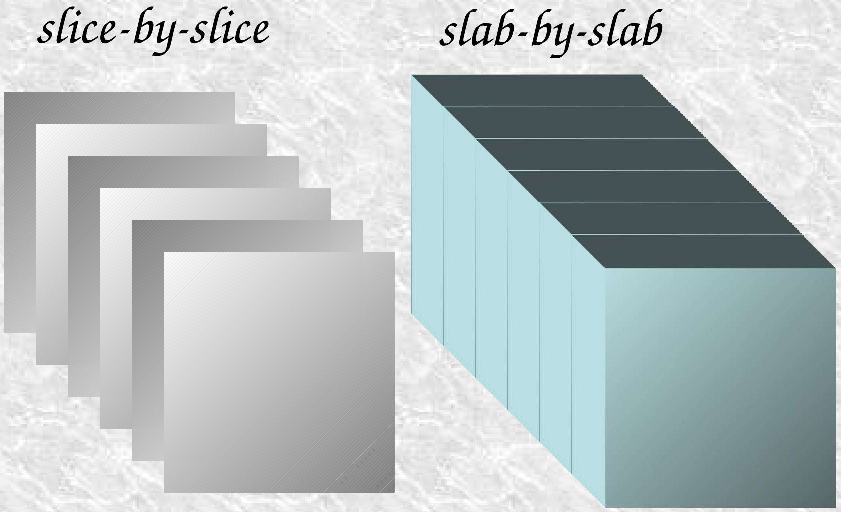
- **Pre-integrate all possible combinations in the TF**



  - One table lookup for each adjacent slices
  - Constant slice sampling

Slice vs Slab

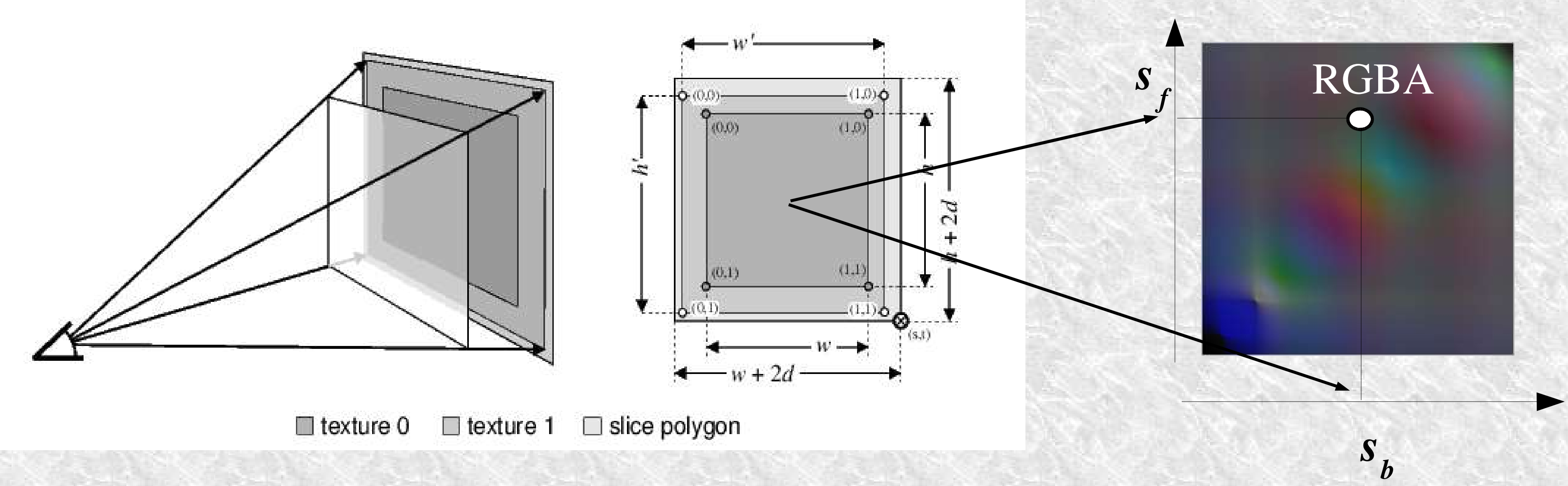
- **Render slab-by-slab instead of slice-by-slice**



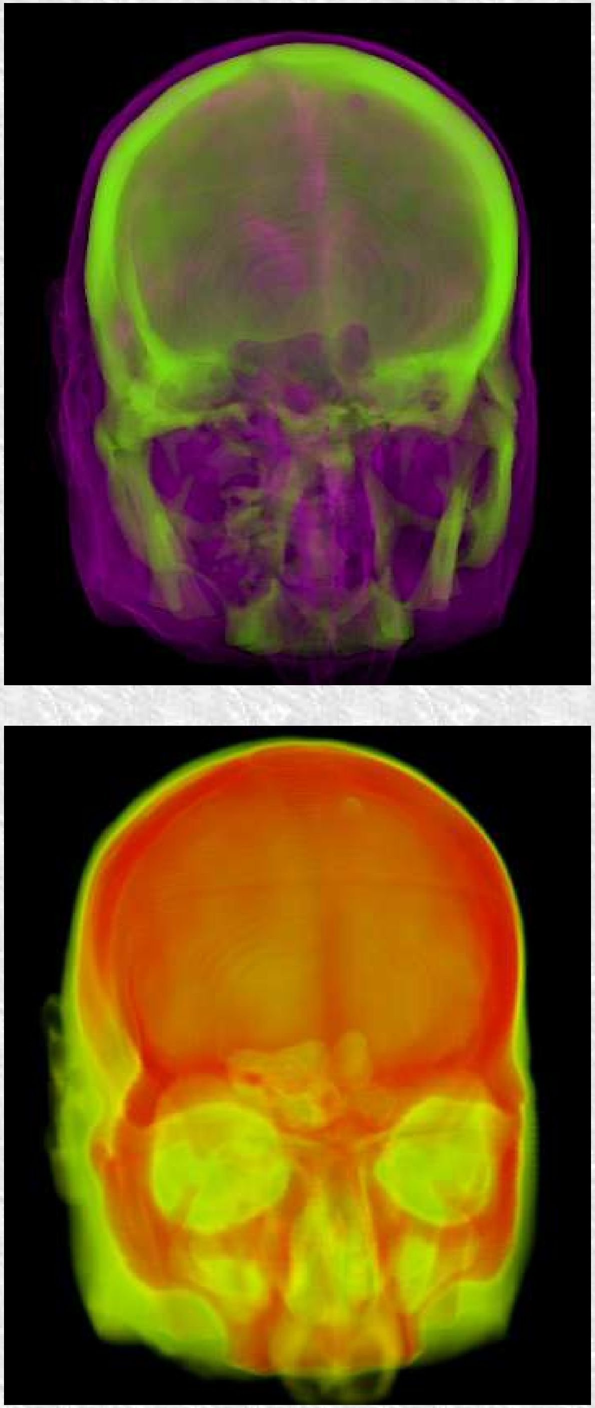
Slab = two adjacent slices

Pre-Integrated Volume Rendering

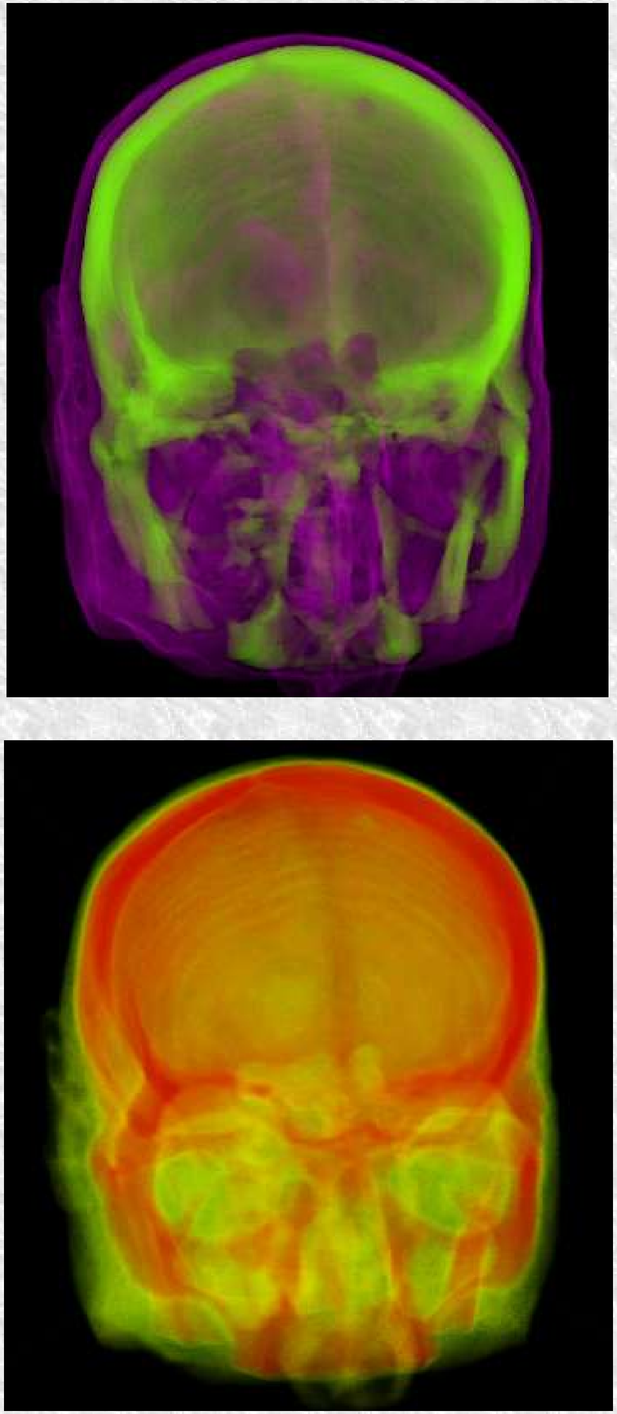
- **Texture fetch**
  - Each slab – 2 textures
  - Use multitextures, adapt texture coordinates for projected slice
  - Dependent texture fetch needed



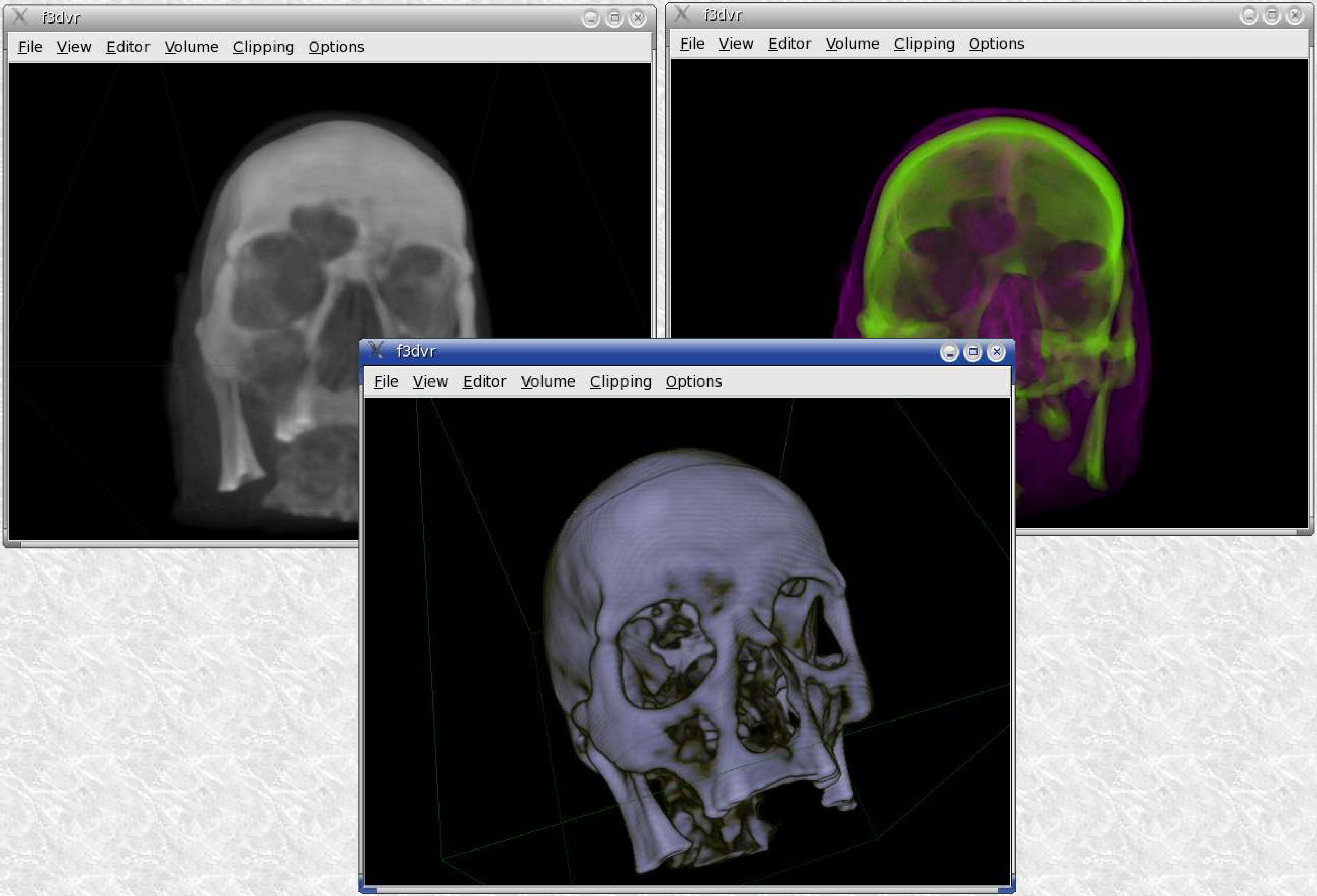
Post-Classification 210 polygons



Pre-Integrated 70 polygons



Skull : Different transfer functions


f3dvr
File View Editor Volume Clipping Options
f3dvr
File View Editor Volume Clipping Options
f3dvr
File View Editor Volume Clipping Options