Fast Visualization of Object Contours by Non-Photorealistic Volume Rendering

Balázs Csébfalvi, Lukas Mroz, Helwig Hauser, Andreas König, Eduard Gröller

Institute of Computer Graphics and Algorithms
Vienna University of Technology

Motivation

- Visualize preferably all the internal structures at the same time
- No a priori knowledge about the data
- Avoid time-consuming specification of an appropriate transfer function
- Interactive frame rates on a low-end PC

Traditional methods

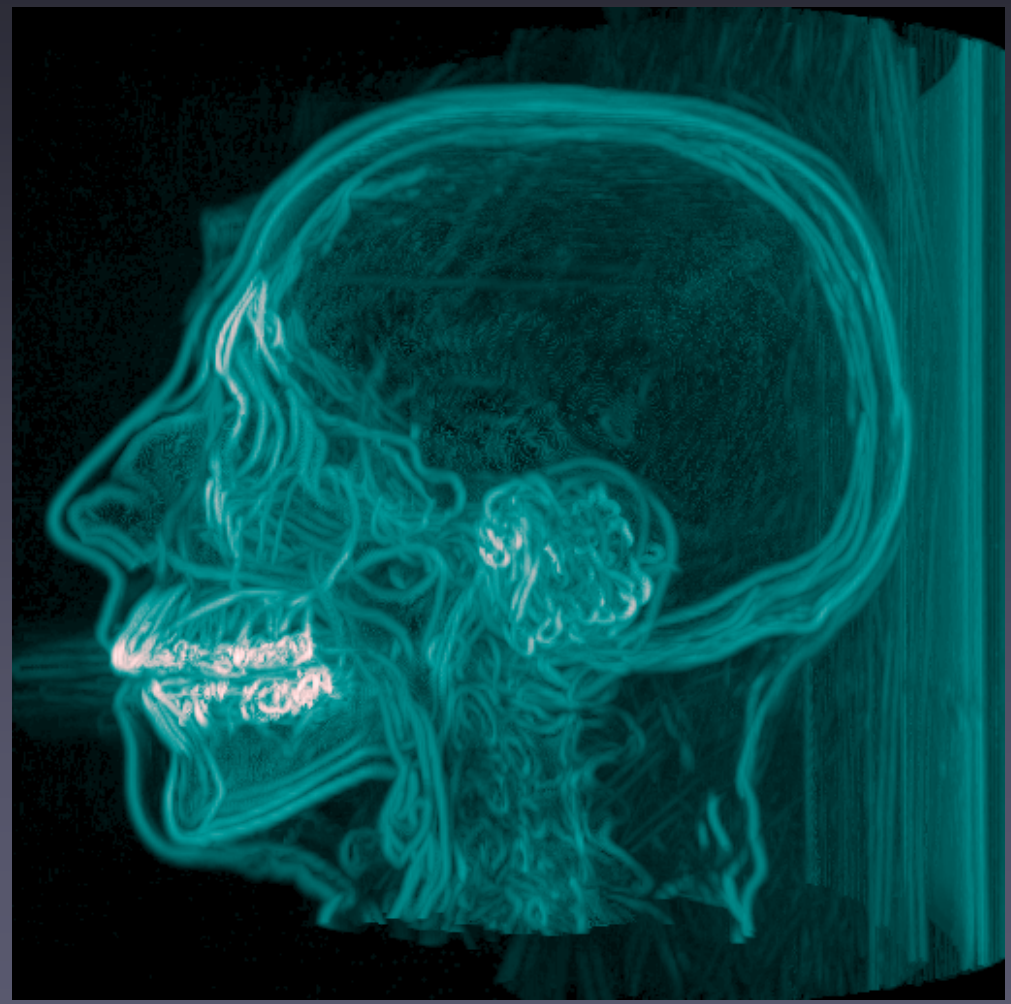
- Indirect volume rendering - “Marching Cubes”
  - computationally expensive preprocessing
  - which iso-surface represents the data best?
  - 3D graphics HW is required for fast rendering
- Direct volume rendering
  - time-demanding transfer function specification
  - HW acceleration is required for fast rendering

Non-photorealistic volume rendering (NPVR)

- Real-Time Volume Previewing [Saito, 1994]
- Principal Direction-Driven 3D LIC [Interrante, 1997]
- Pen-and-Ink Rendering in Volume Visualization [Treavett & Chen, 2000]
- Volume Illustration: NPR of Volume Data [Ebert & Rheingans, 2000]

Contour projection

- Emphasize the surface voxels (the gradient magnitude represents the “surfaceness”)
- View-dependent intensity calculation - contour enhancement
- Maximum intensity projection (MIP)



Contour enhancement

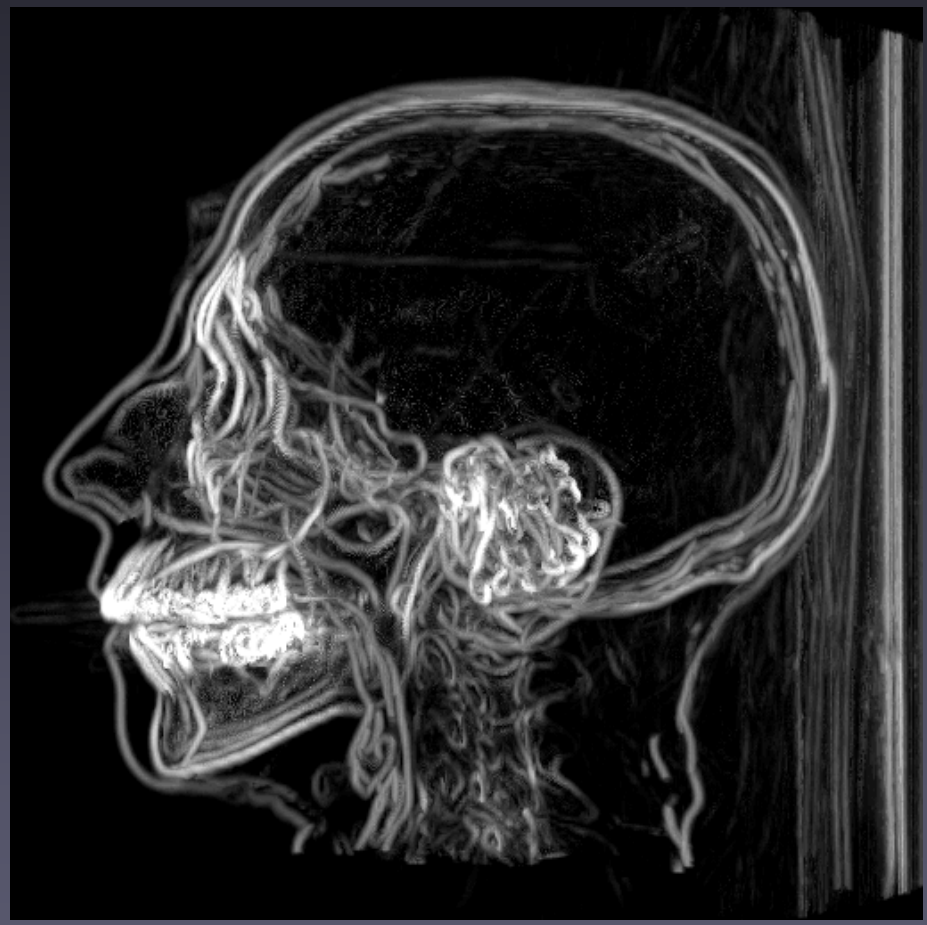
- View-dependent intensity function:

$$I(\mathbf{p},\mathbf{v}) = (1 - \mathbf{v} \cdot \frac{\nabla f(\mathbf{p})}{|\nabla f(\mathbf{p})|})^n \cdot g(|\nabla f(\mathbf{p})|)$$

  - n controls the sharpness
  - $g(|\nabla f(\mathbf{p})|)$ is a windowing function
- In order to keep the depth information:
  - Depth cueing
  - Local maximum intensity projection (LMIP)
  - Alpha-blending compositing, where color: $g(|\nabla f(\mathbf{p})|)$, opacity: $I(\mathbf{p},\mathbf{v})$

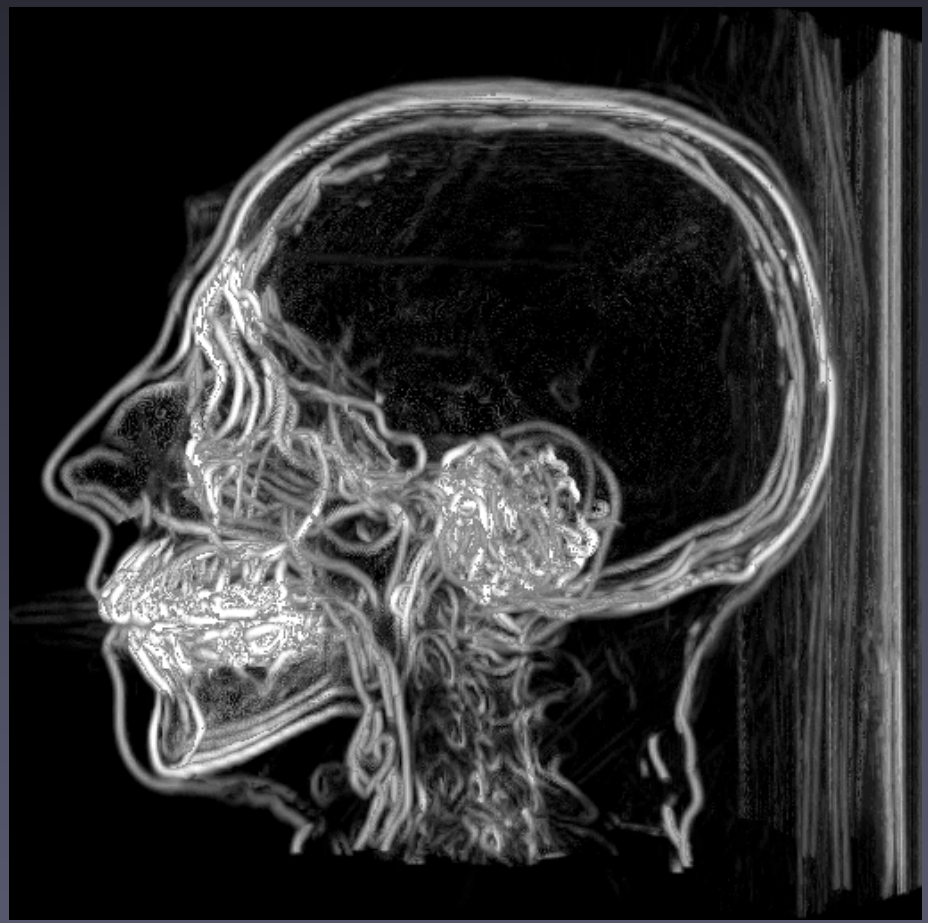
Variations of contour projection

Maximum Intensity Projection (MIP)



MIP using depth cueing

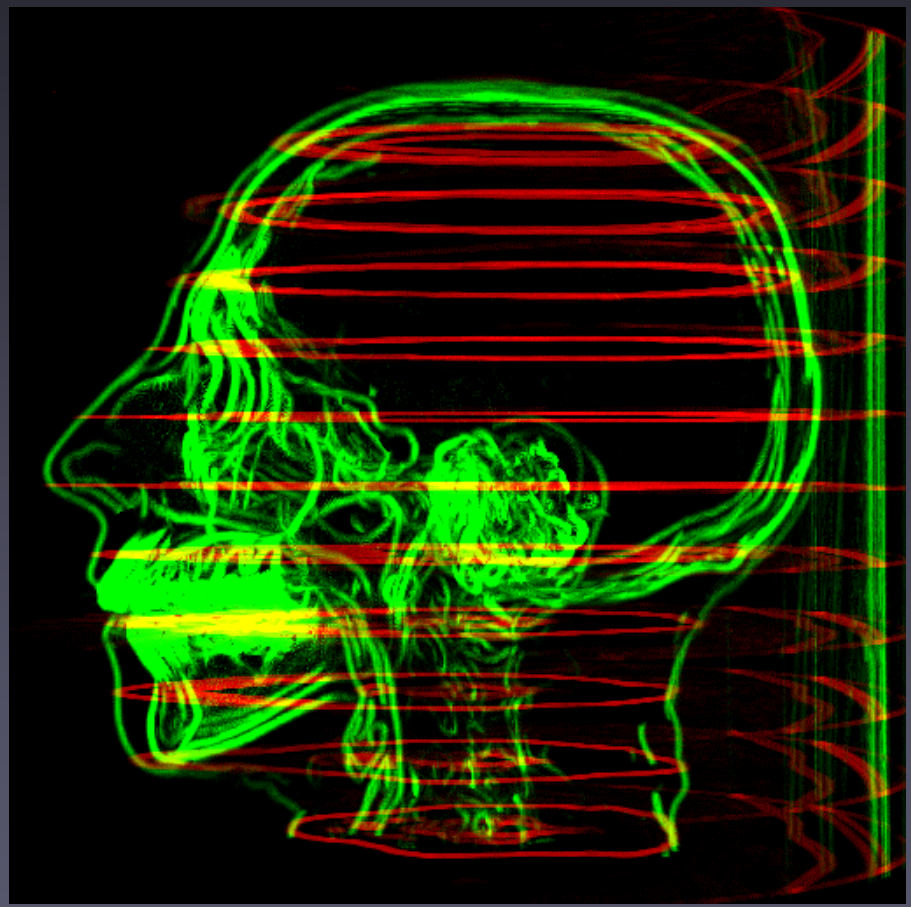
Variations of contour projection



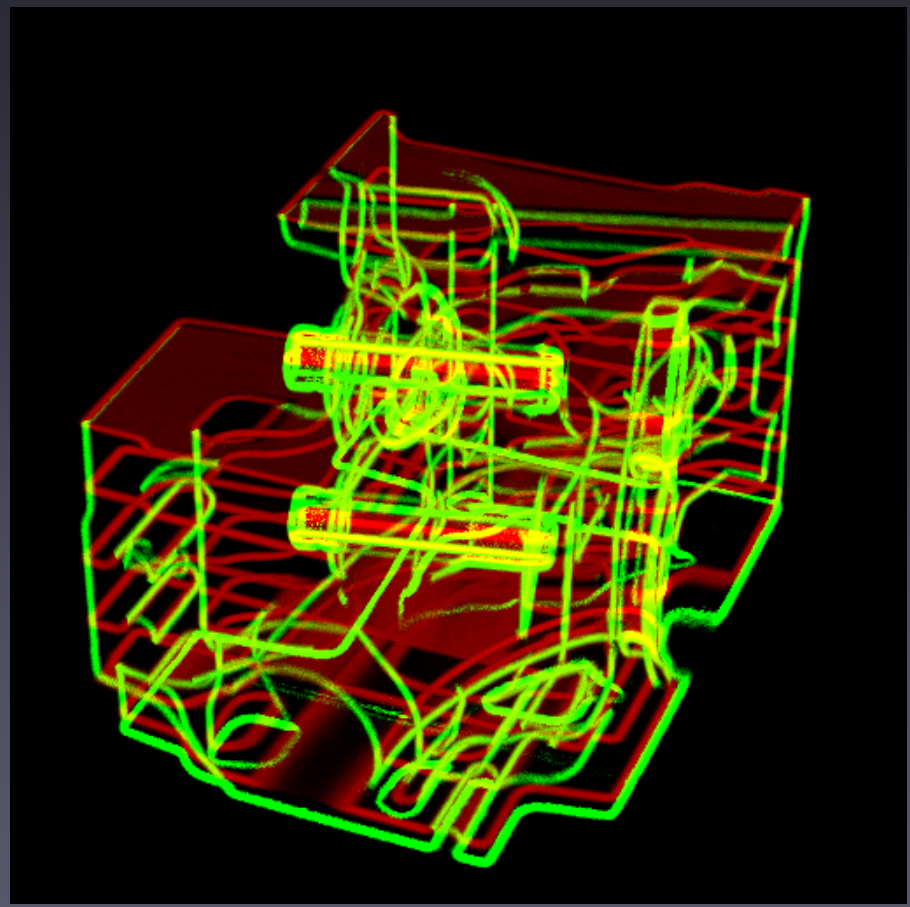
Local Maximum Intensity Projection

Alpha-blending compositing

Adding level lines

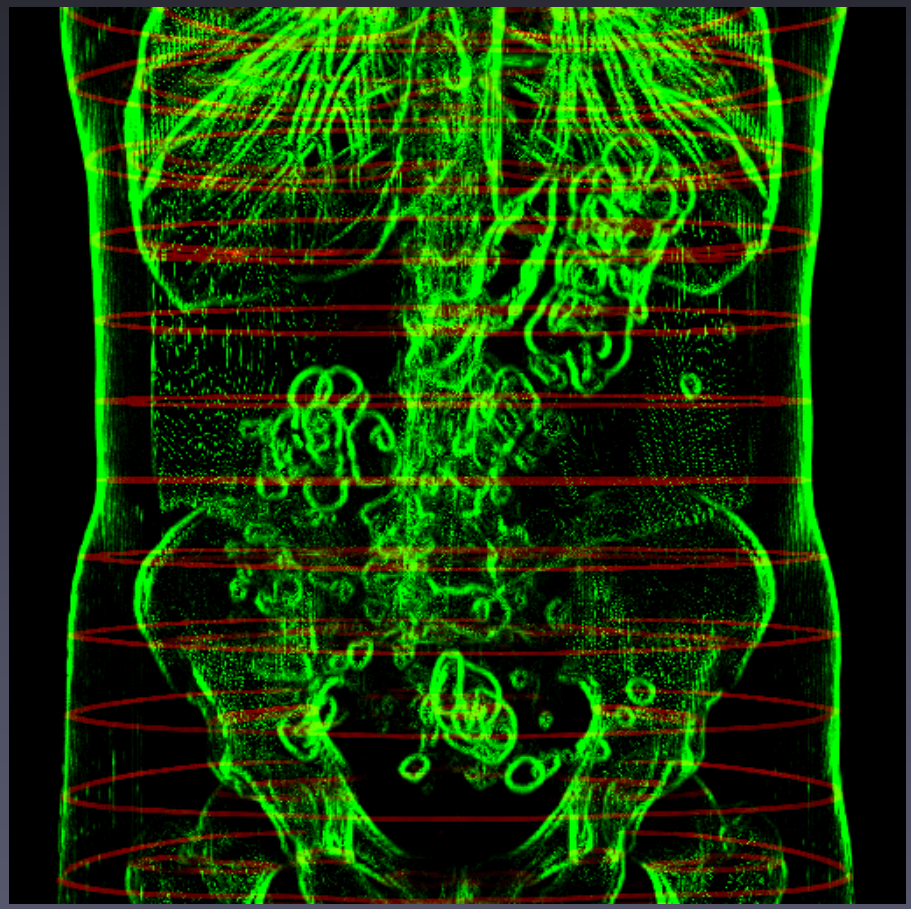


CT head with level lines

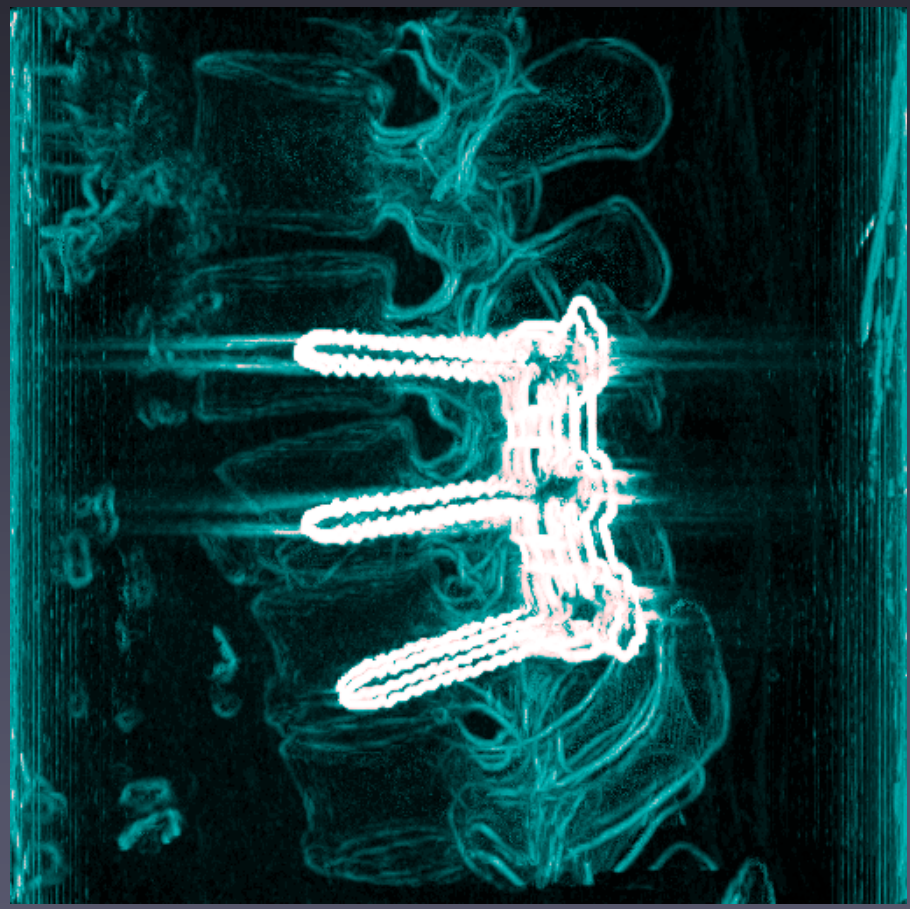


Engine block with level lines

Visualization of internal structures



Human lungs with the bronchia



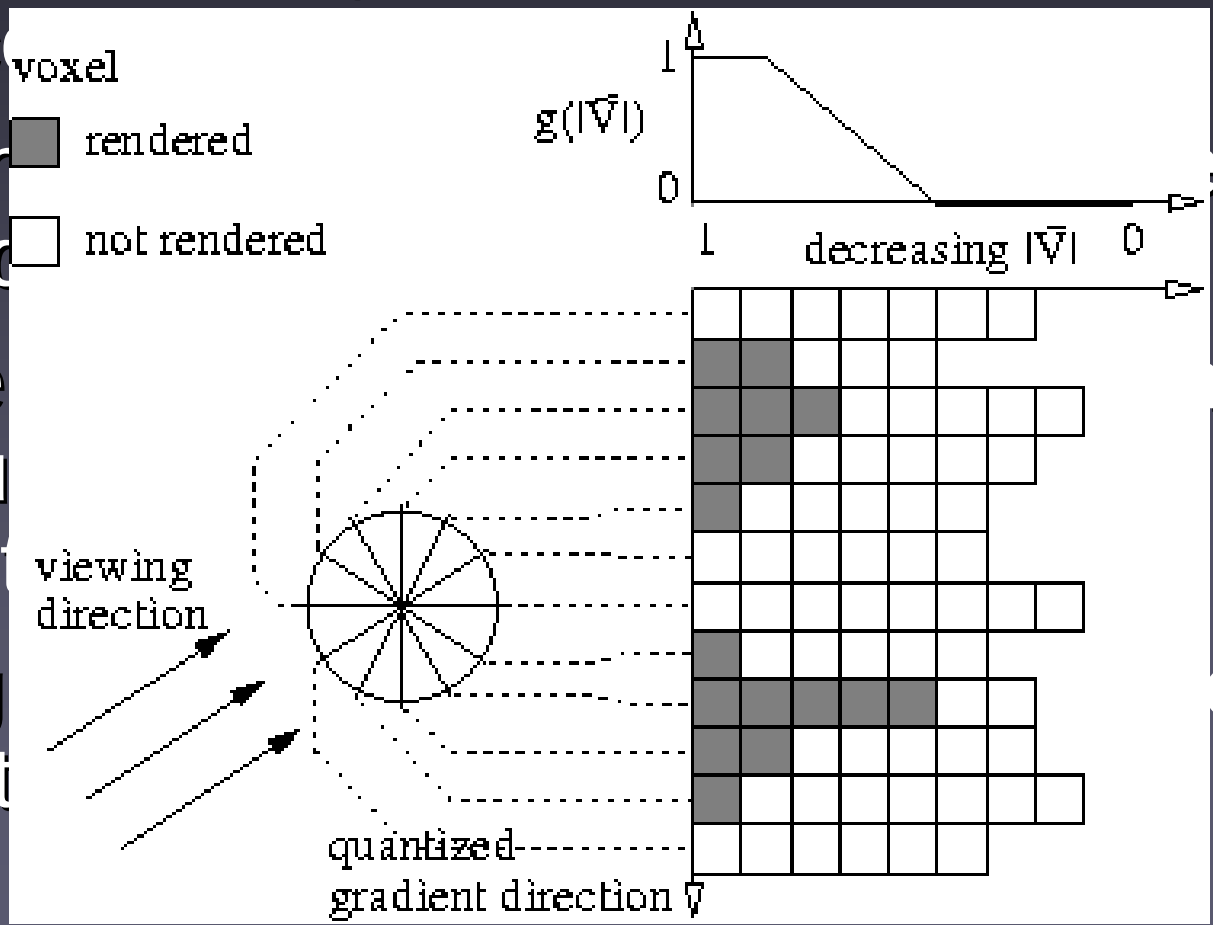
Human vertebrae with screws

How to make it interactive?

- Gradient quantization (12 bit representation)
- View-dependent components of the intensity function are stored in a LUT
- Preprocessing - voxels are sorted according to the view-independent component
- Fast rendering using an efficient shear-warp projection

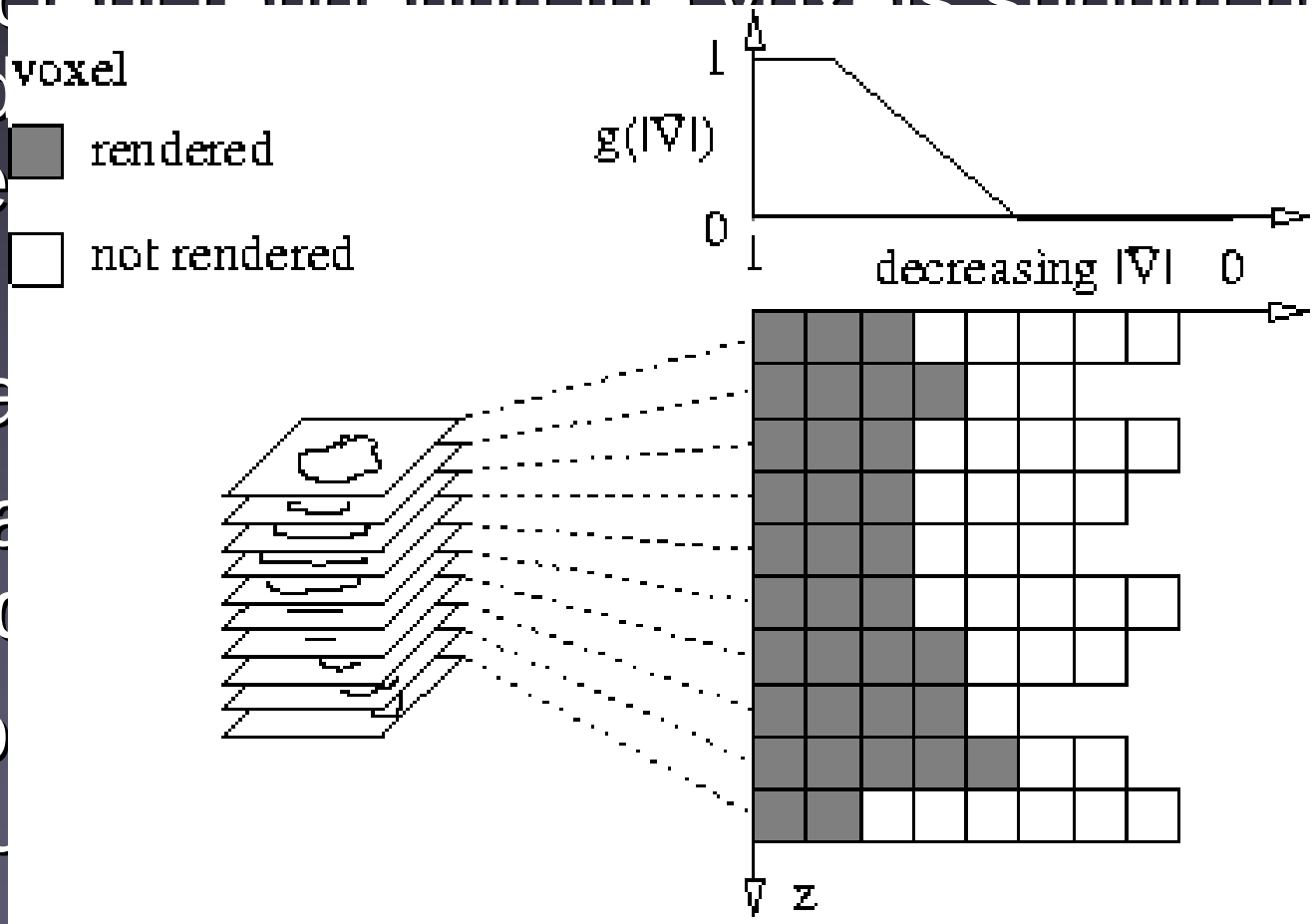
Preprocessing for MIP

- For efficient rendering the voxels should be ordered acc
- Global sortir
  view-depend
- Decompose
  into finite nu
  grouped by
- Skip the reg
  because of t
  factor



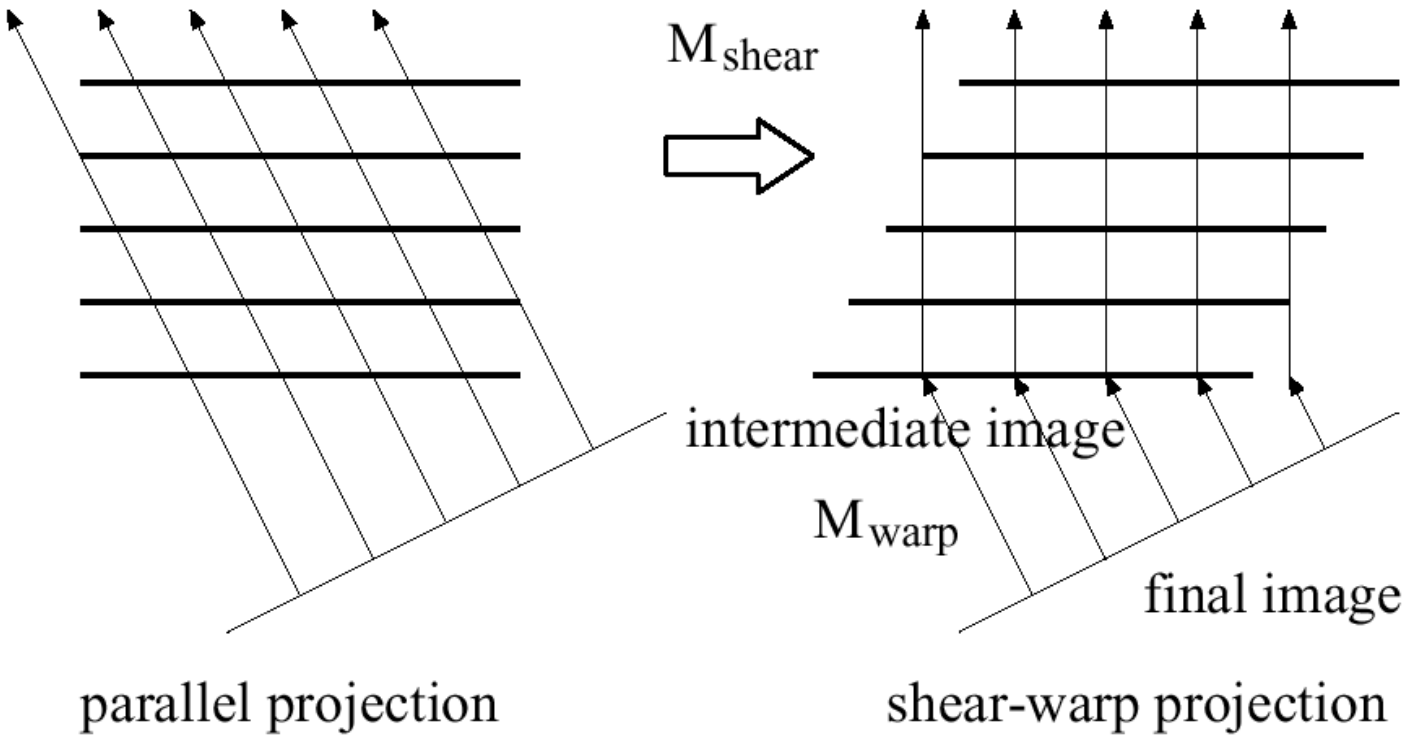
Preprocessing for alpha-blending compositing

- For efficient rendering the voxels should be ordered
- Voxels
- Within a
- Skip ho

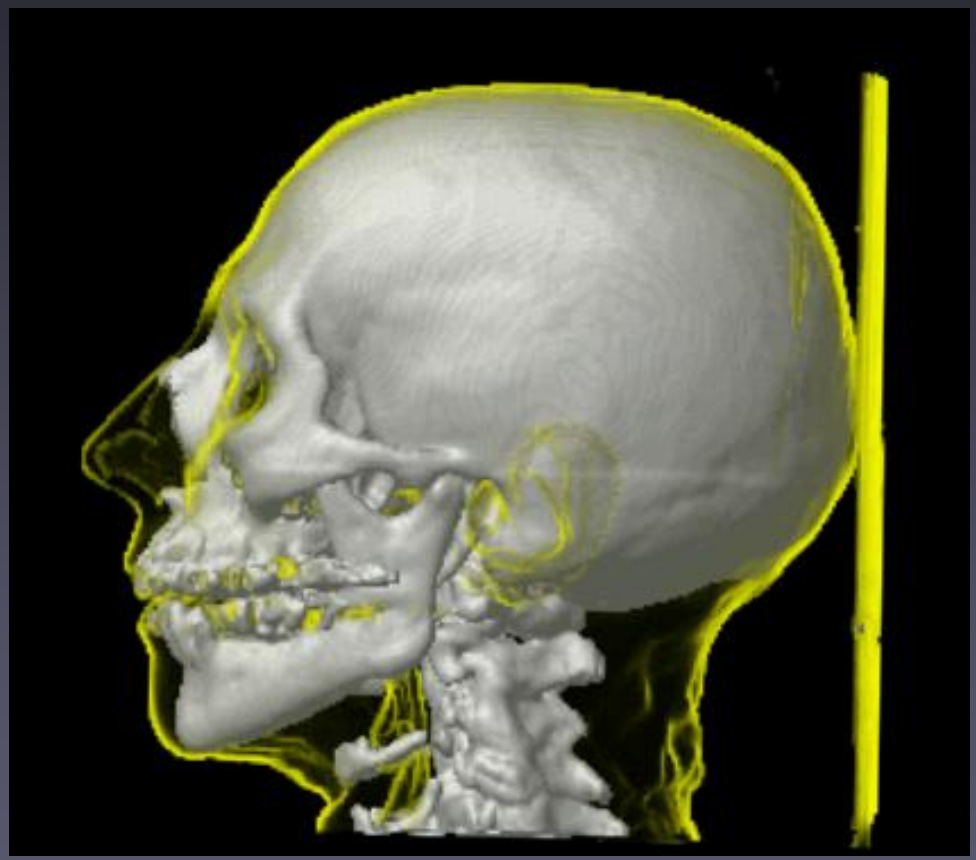


Shear-warp projection

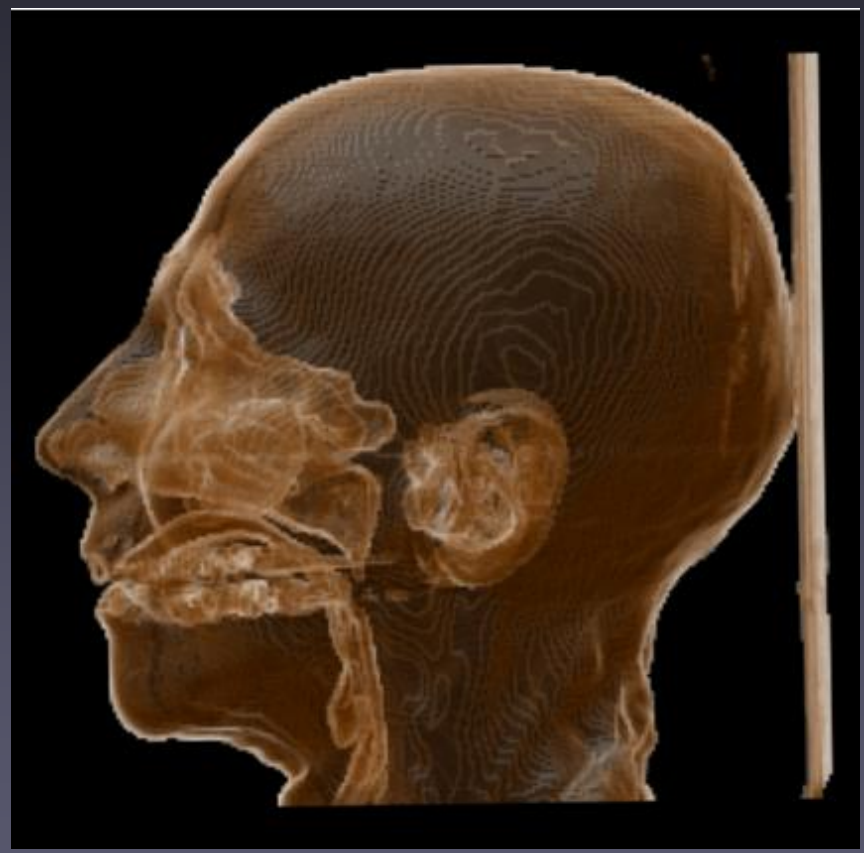
$$M_{view} = M_{warp} \cdot M_{shear}$$



Combination with traditional volume rendering



Combination with a shaded isosurface



Combination with Phong shading

Rendering times

Volume	Head		Vertebrae	
Resolution	256 × 256 × 225		256 × 256 × 241	
Model	MIP	Blending	MIP	Blending
Rendered voxels	102k	366k	337k	942k
Rendering time	85ms	150ms	130ms	270ms

Conclusion

- Visualization of internal structures without transfer function specification
- The intensity function does not directly depend on the densities $\Rightarrow$ no a priori knowledge about the data is required
- Simplified NPR model $\Rightarrow$ data reduction
  - no visual overload
  - interactive rendering even on a low-end PC