
vvis
volume visualization

Overview

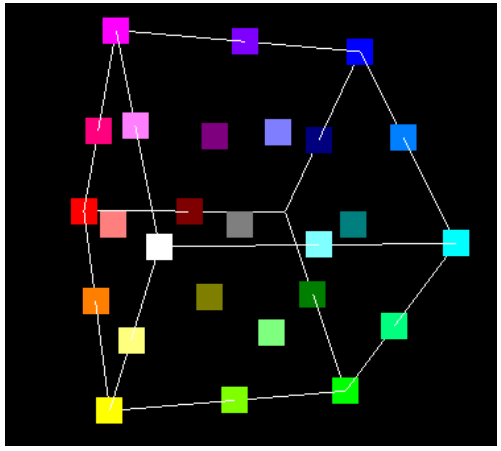
1. **Volume data, data formats and transfer function**
2. Implemented renderer
3. Real time example classification of an engine

Volume data

- mathematical definition

$$\phi : \mathbb{R}^3 \mapsto \mathbb{R}$$
$$\phi(x, y, z)$$

- data of discrete volume



Example: Computer tomography

from: http://de.wikipedia.org/wiki/Bild:Sensation16.JPG

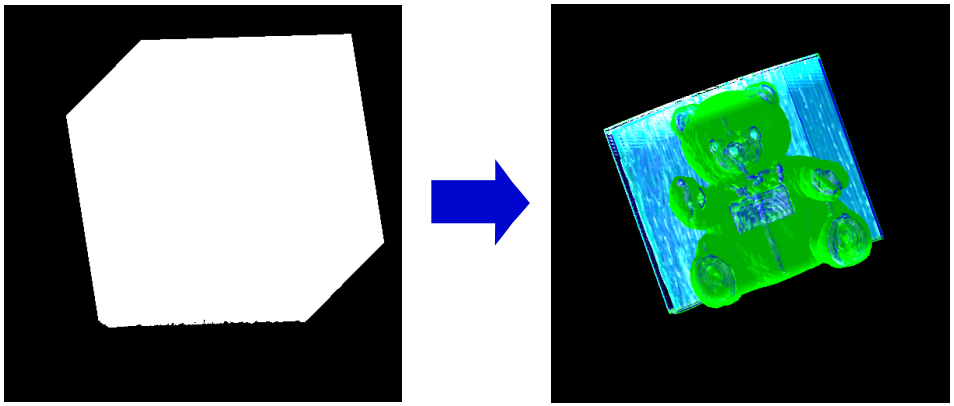
density values: 0-255 (8Bit)

data formats

- OpenQVis (University of Erlangen)
- Vrend (IWR Heidelberg)
- series of tiff images

transfer function (I/II)

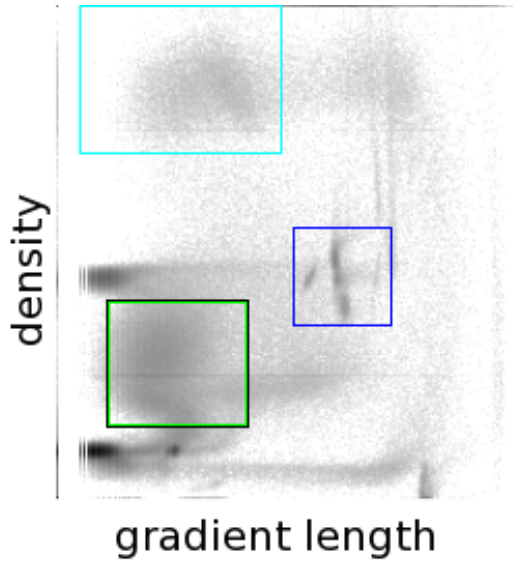
- Definition
  $TF : (density, gradientlength) \mapsto (colors, extinction)$
- Why do we need it?



transfer function (II/II)

- **Volume data classification**
  Each density value / gradient length is assigned the following properties:

  - emissivity
  - absorptivity
  - diffuse
  - specular
  - glossiness



Overview

1. Volume data, data formats and transfer function
2. **Implemented renderer**
3. Real time example classification of an engine

Available renderers

1. Texture based
2. Raycasting
3. Shear Warp

Renderer - Raycasting

- Image order algorithm
- Scalable output size
- Uses all information provided by the transfer function
- Good quality
- Relatively slow

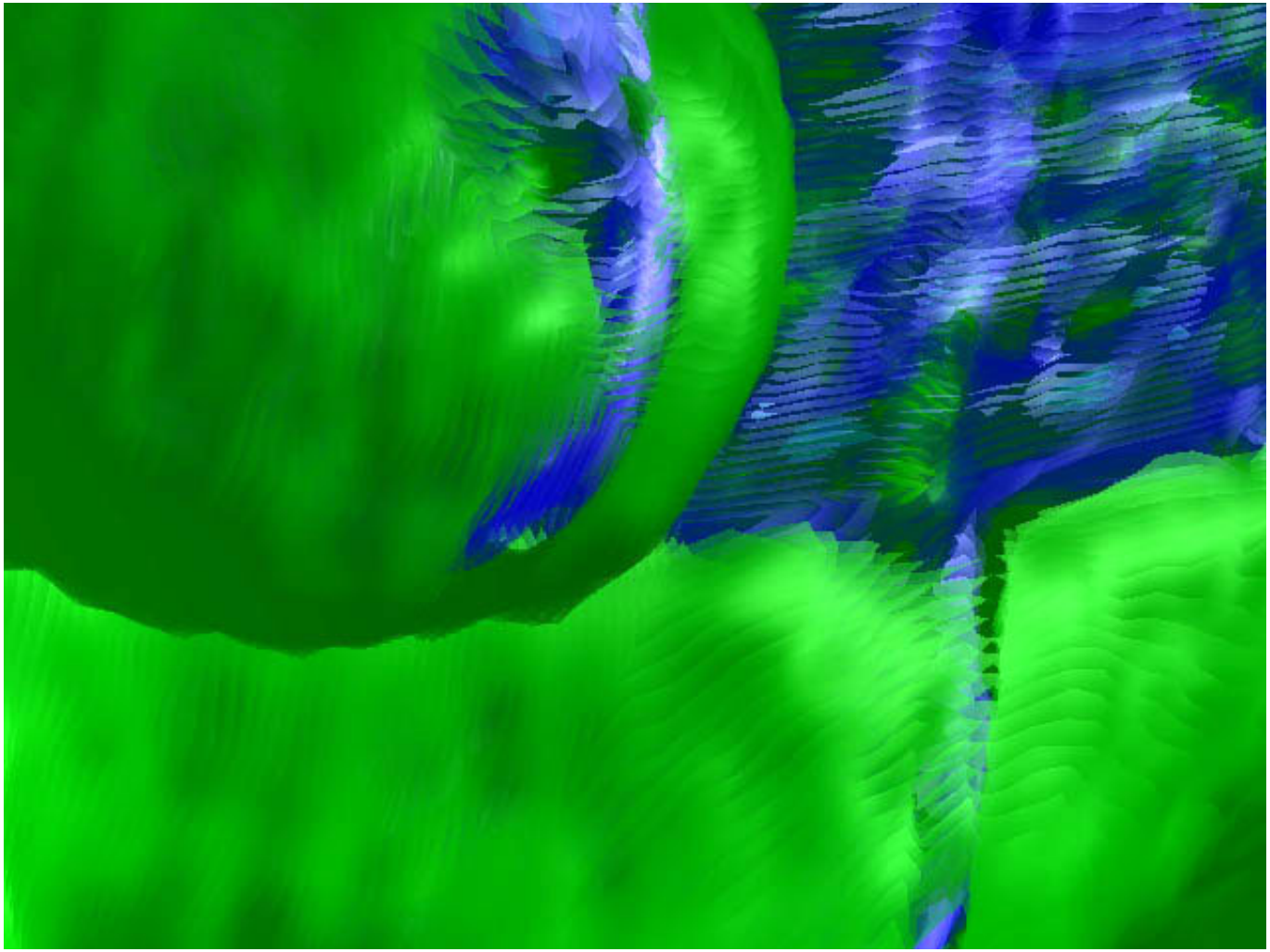
Renderer - Raycasting

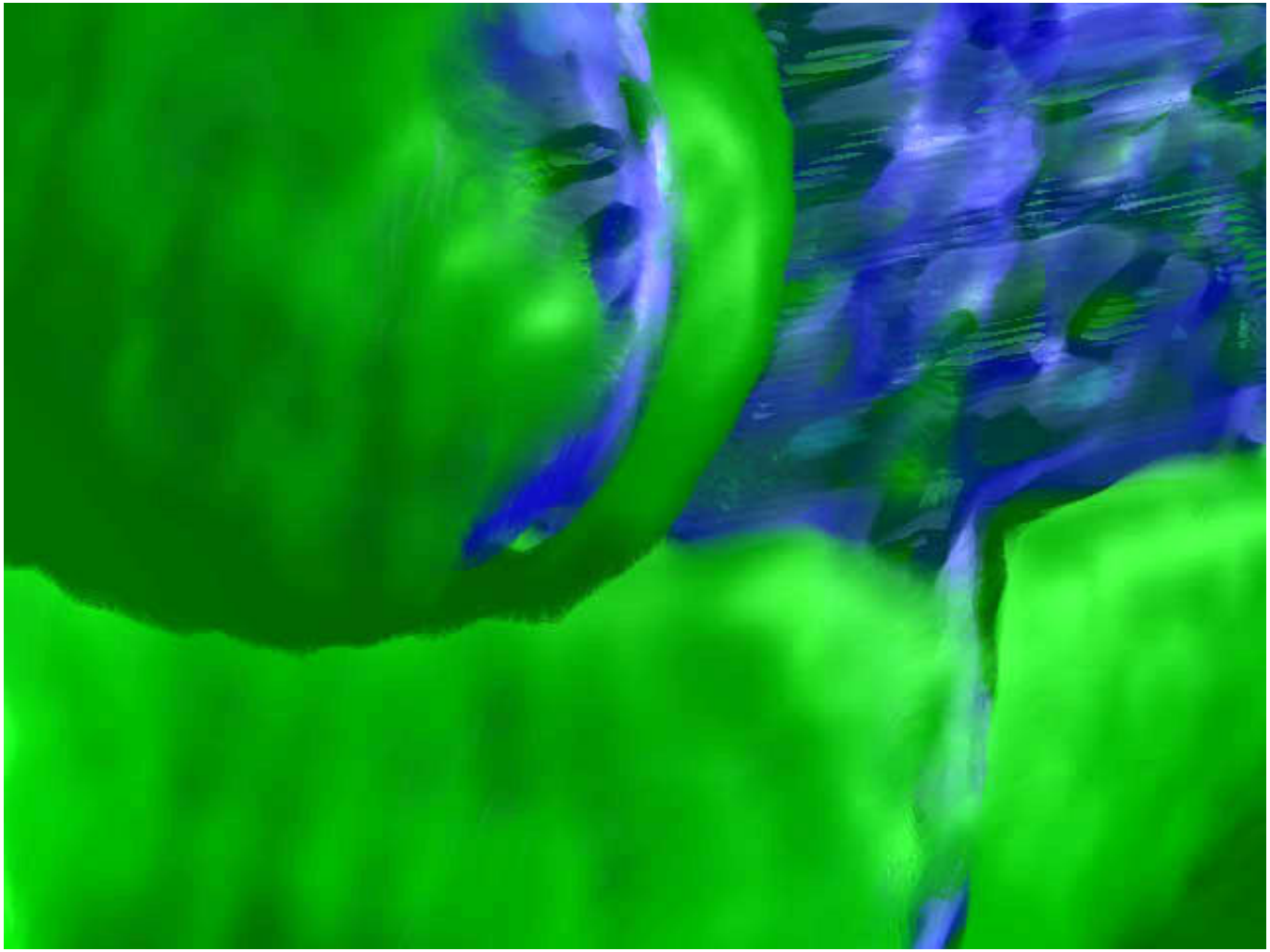
$$
\begin{aligned}
I &= \int_0^D color(\vec{x}(\lambda)) \exp\Big(- \int_0^\lambda \mathbf{extinction}(\vec{x}(\mu)) d\mu \Big) d\lambda \\
color(\vec{x}) &= \mathbf{Emissivity}(\vec{x}) + Diffuse(\vec{x}) + Specular(\vec{x}) \\
Diffuse(\vec{x}) &= (\mathrm{normal}(\vec{x}), \mathrm{LightDirection}) \mathrm{LightColor} * \mathbf{DiffuseColor}(\vec{x}) \\
Specular(\vec{x}) &= (\mathrm{normal}(\vec{x}), \mathrm{LightDirection})^{\mathbf{Glossiness}(\vec{x})} \\
&\quad \mathrm{LightColor} * \mathbf{SpecularColor}(\vec{x})
\end{aligned}
$$

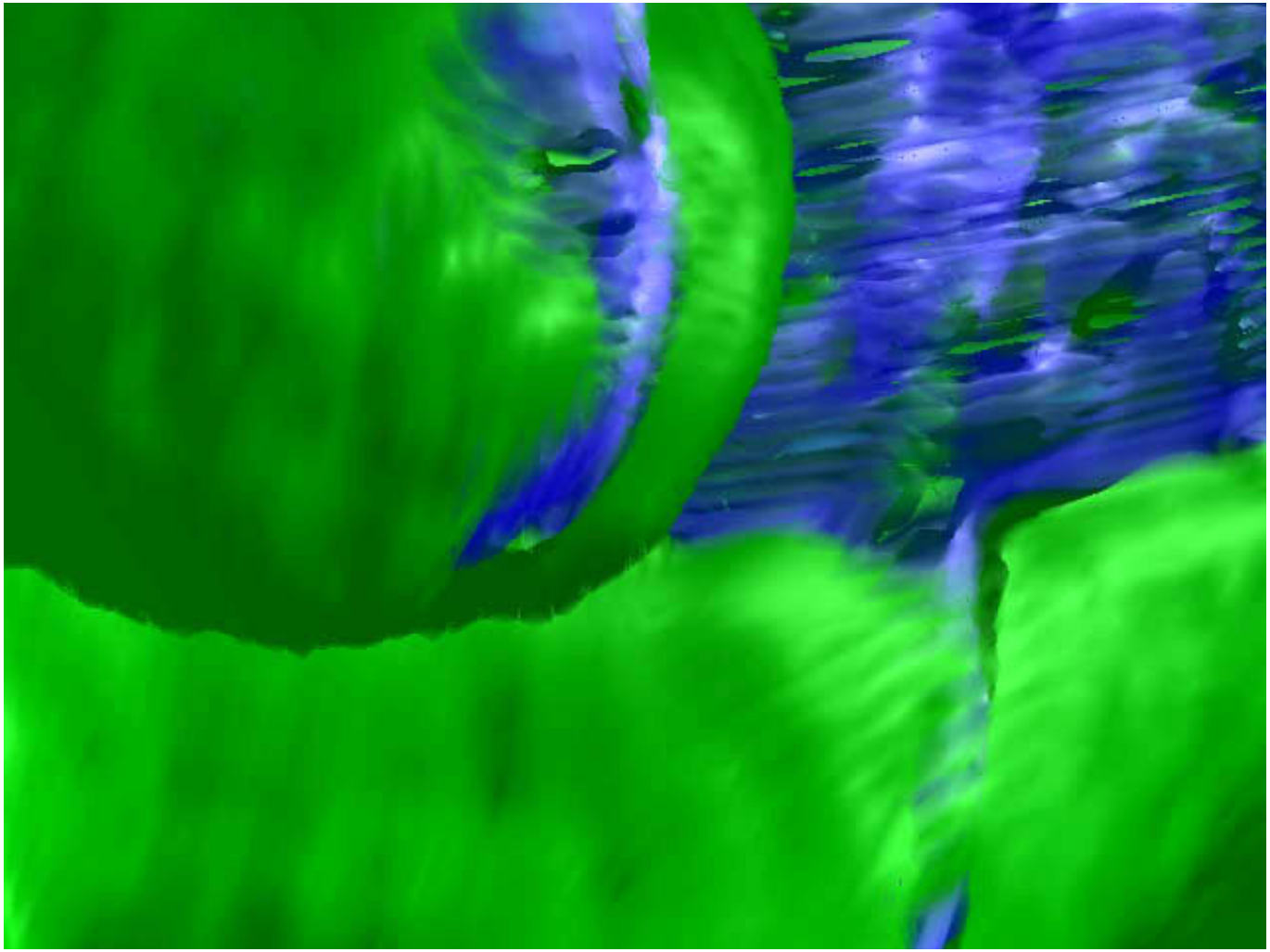
Renderer - Raycasting - Pre-Integration

- Idea from hardware rendering techniques
- Improves quality for 'rough' transfer functions
- Linear interpolation of voxel and gradient
  length from start to end of integration
  step interval before classification
- No lookup table in our implementation due to using gradient length

The following Teddy volume data set has a resolution of 128 x 128 x 62.



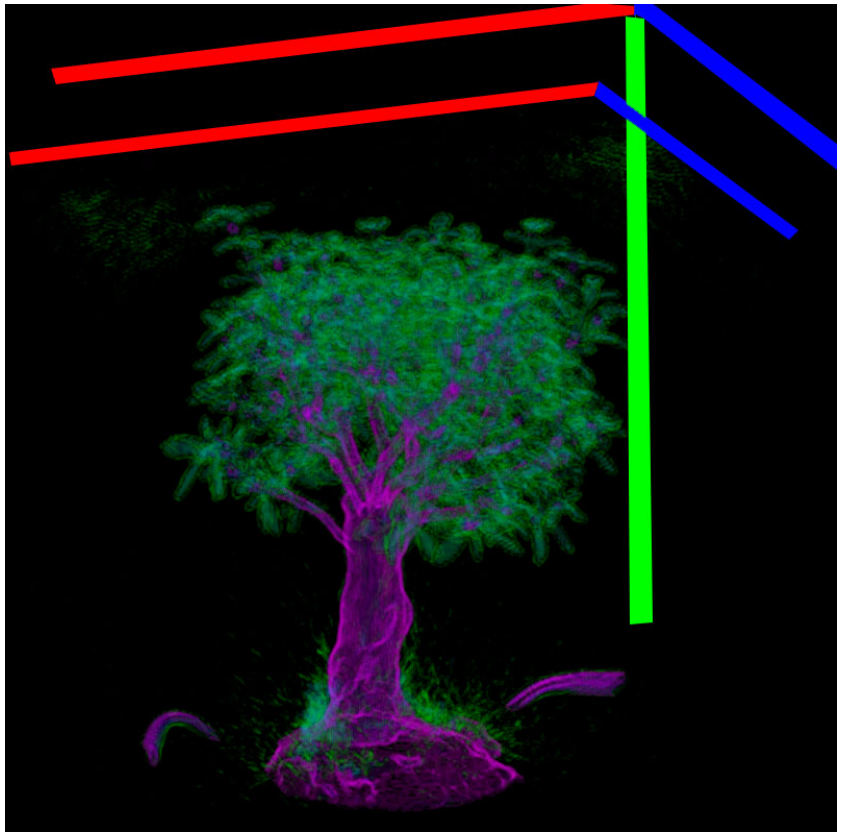




Renderer - Texture based

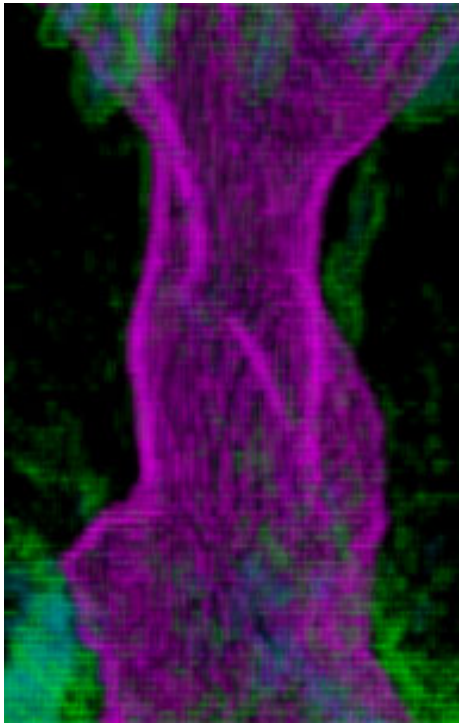
- fast renderer, nice to get a rough overview
- 3 Stacks of 2D RGBA Textures, 2 of them are drawn at a time
- Textures on OpenGL Triangles
- Graphic card's memory is limiting resolution
- No light

Renderer - Texture based

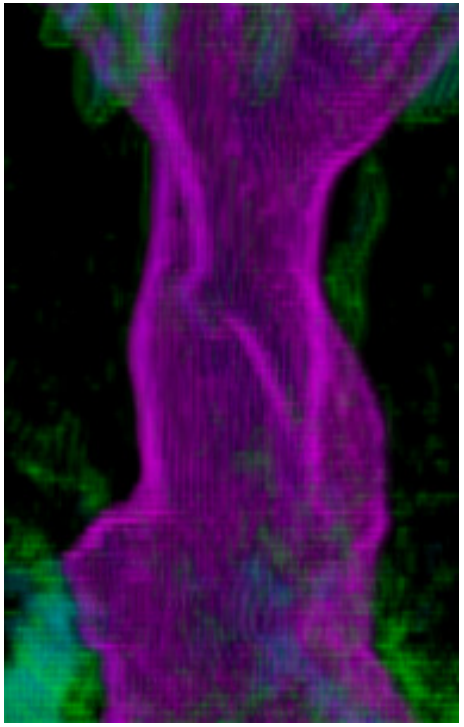


Renderer - Texture based

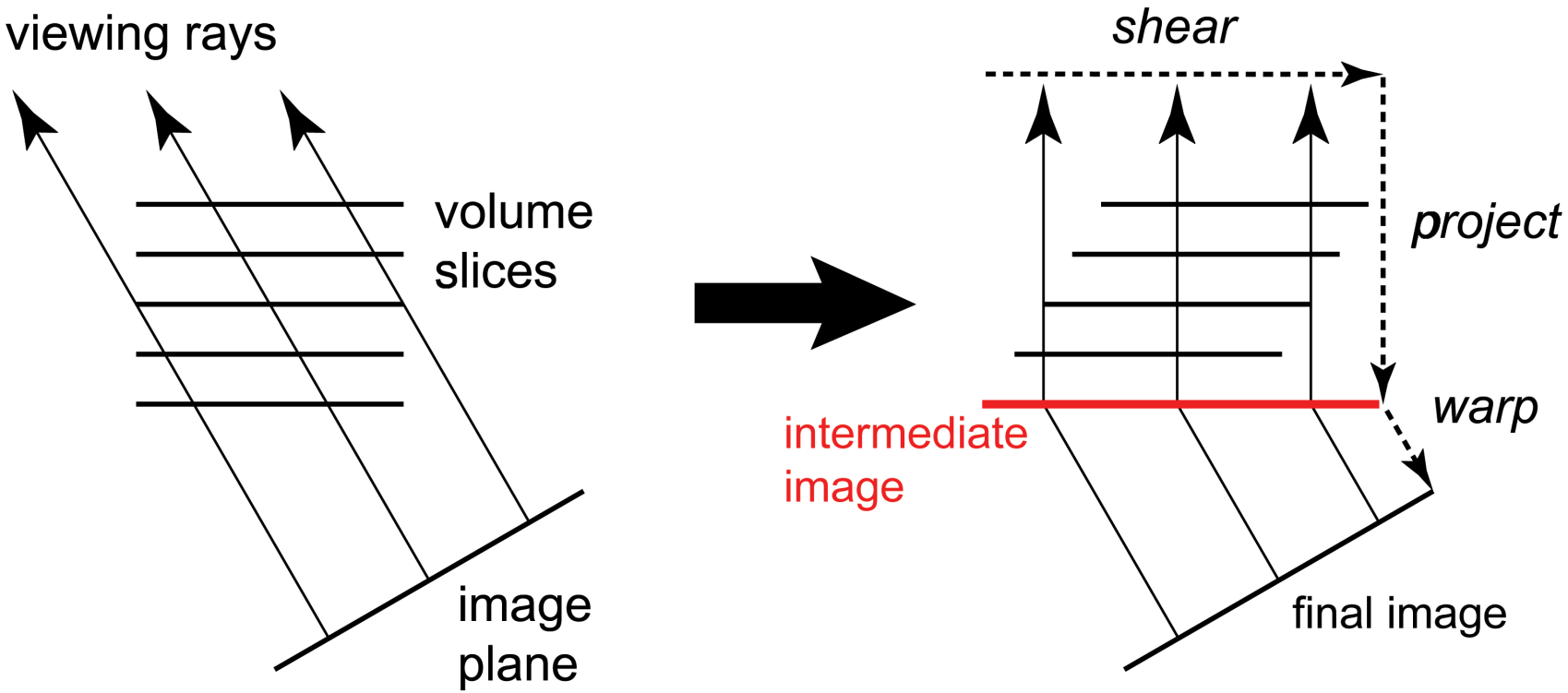
Standard



using pre-integration



Renderer - Shear-Warp



based on image taken from Philippe Lacroute's thesis 'FAST VOLUME RENDERING USING A SHEAR-WARP FACTORIZATION OF THE VIEWING TRANSFORMATION'

Renderer - Shear-Warp

Reasons for speedup compared to raycasting

- Object Order Algorithm
- RLE of classified volume data
- fixed weights for interpolation for a whole plane

The RLE is done once before rendering for a given transfer function. This takes 2-3 seconds on a current hardware for the example data set we want to classify at the end of this presentation.

Overview

1. Volume data, data formats and transfer function
2. Implemented renderer
3. **Real time example classification of an engine**