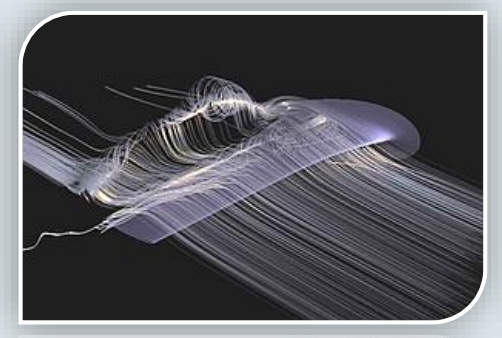
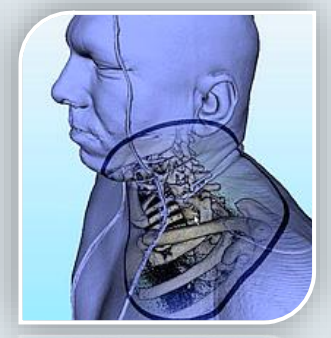
SIEMENS
Ingenuity for life


# Visual Data Analytics
# Volume Visualization

Dr. Johannes Kehrer

# Disclaimer

SIEMENS
Ingenuity for life

These lecture slides and materials are for personal use by students and are intended for educational purposes only. Distribution or reproduction of the material is prohibited.

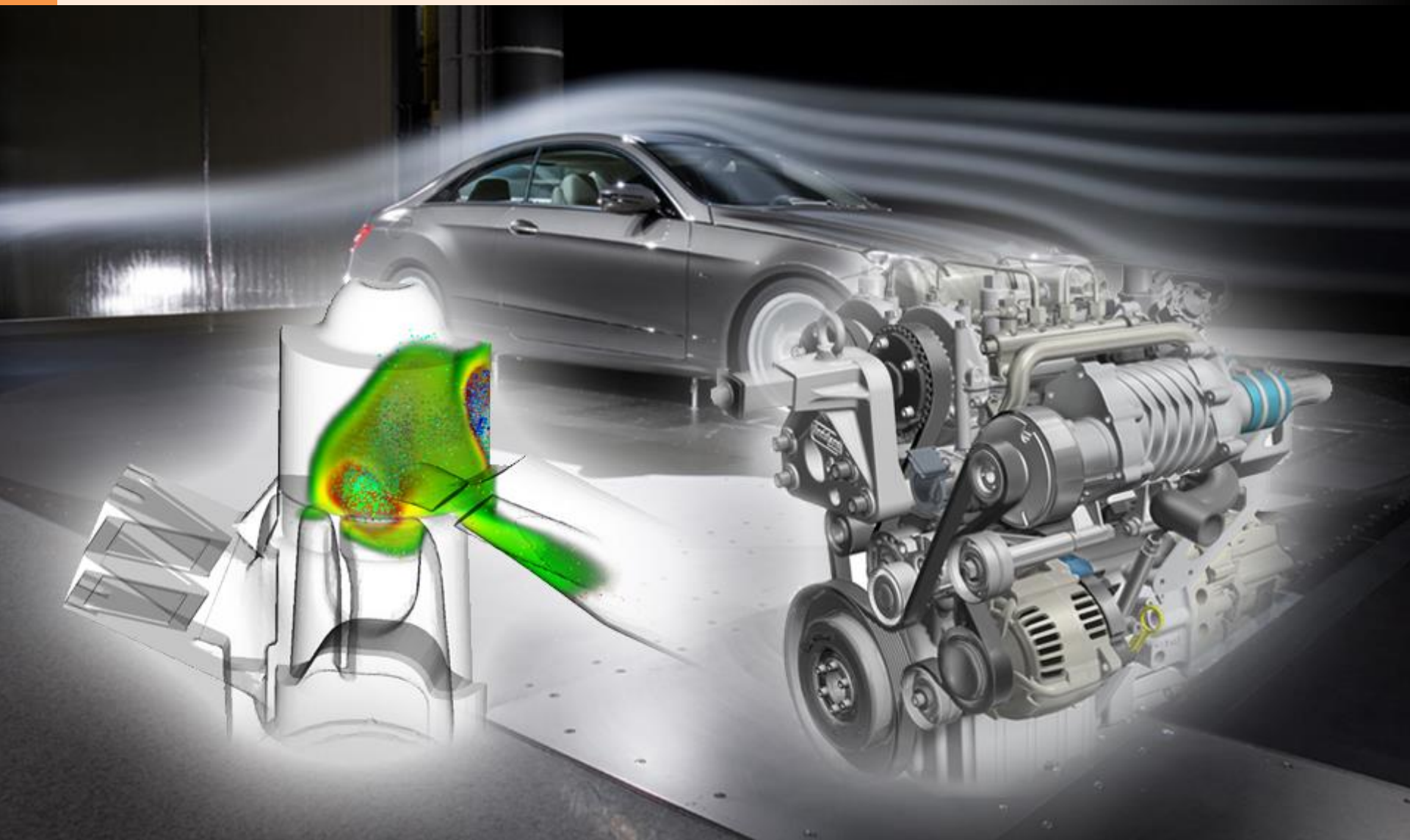
The views and opinions expressed by any individual during the course of these lecturers are their own and do not necessarily represent the views, opinions, or positions of Siemens or the Technical University of Munich.

Although reasonable efforts have been made to ensure the information in these lecture slides is current and accurate, we do not assume any responsibility for the content of the posted material.

These slides also contain links to websites of third parties. As the content of these websites is not under our control, we cannot assume any liability for such external content.

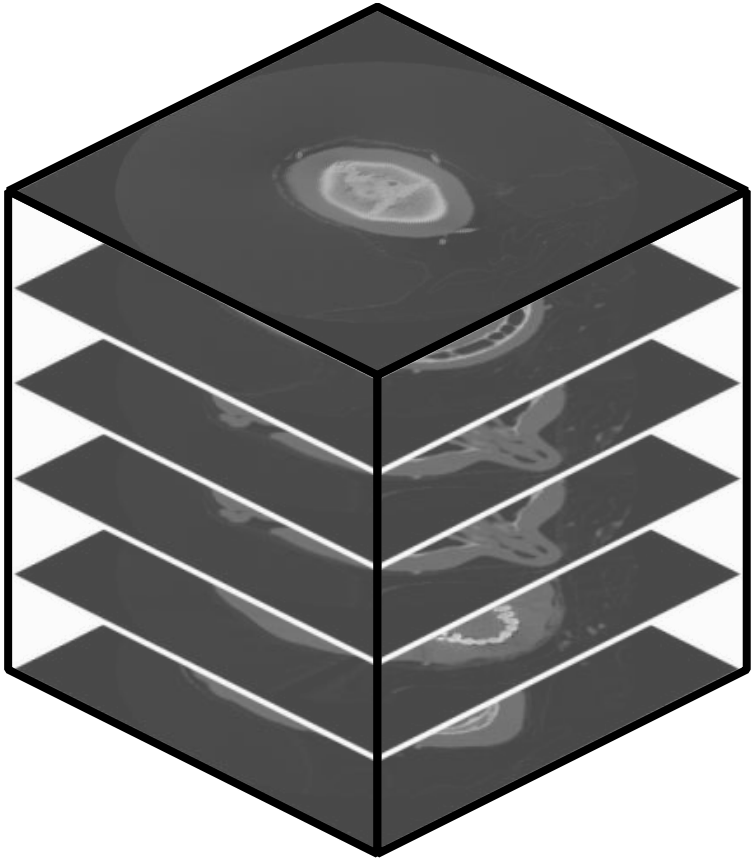
# Medical scanners

# Automotive Engineering



# Volume visualization

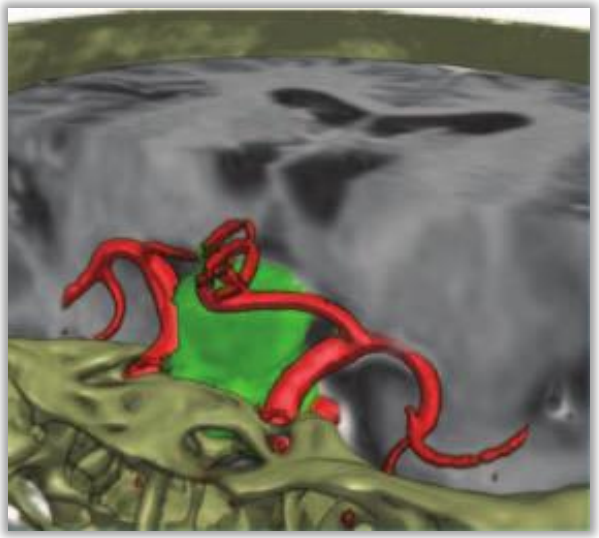
SIEMENS
Ingenuity for life

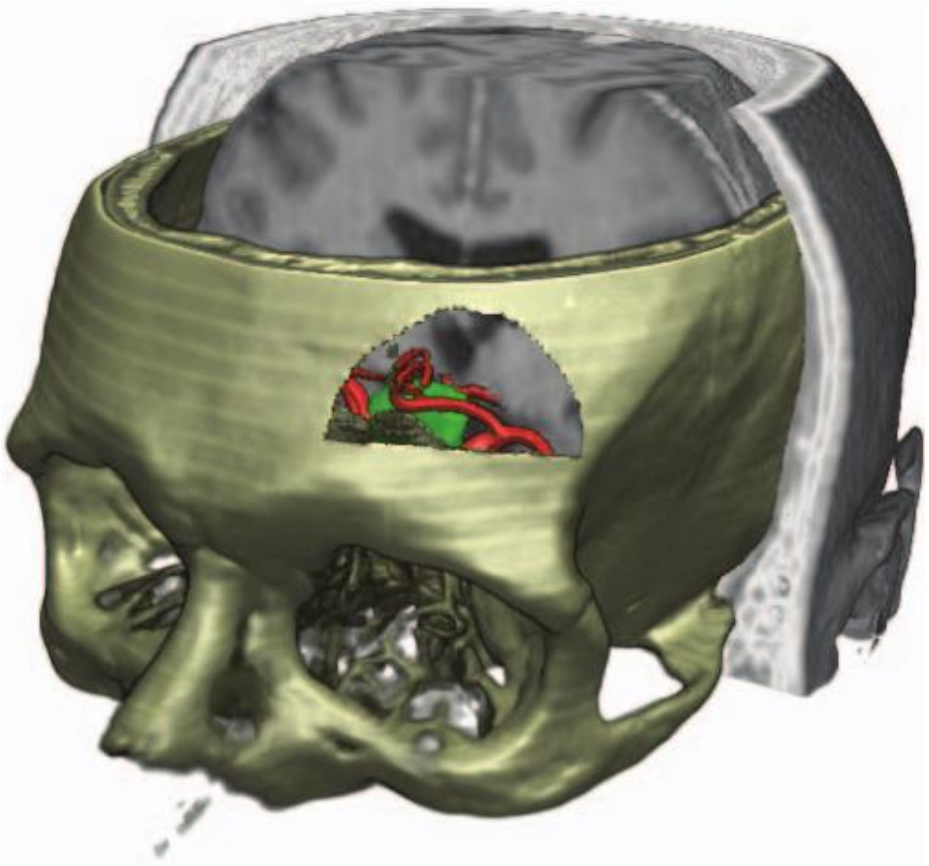


# Visualization – Examples

SIEMENS
Ingenuity for life

- Preoperative planning of a tumor resection





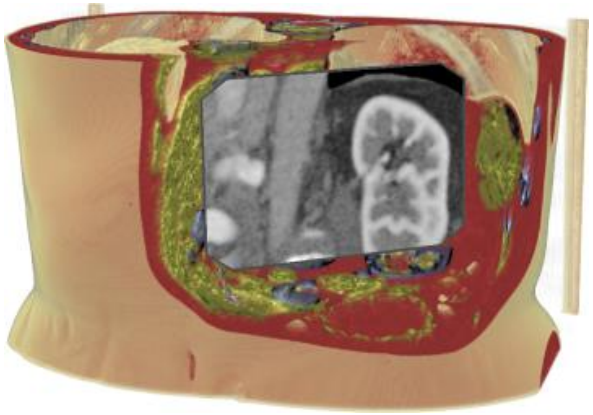
[Beyer et al. 2007]

Black/white: brain – Magnetic Resonance Imaging (MRI)
Green: tumor – MRI
Red: vessels – Magnetic Resonance Angiogram (MRA)
Brown: skull – Computer Tomography (CT)

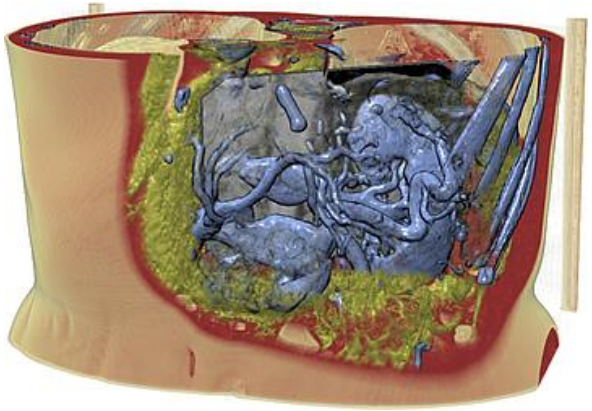
# Visualization – Examples

SIEMENS
Ingenuity for life

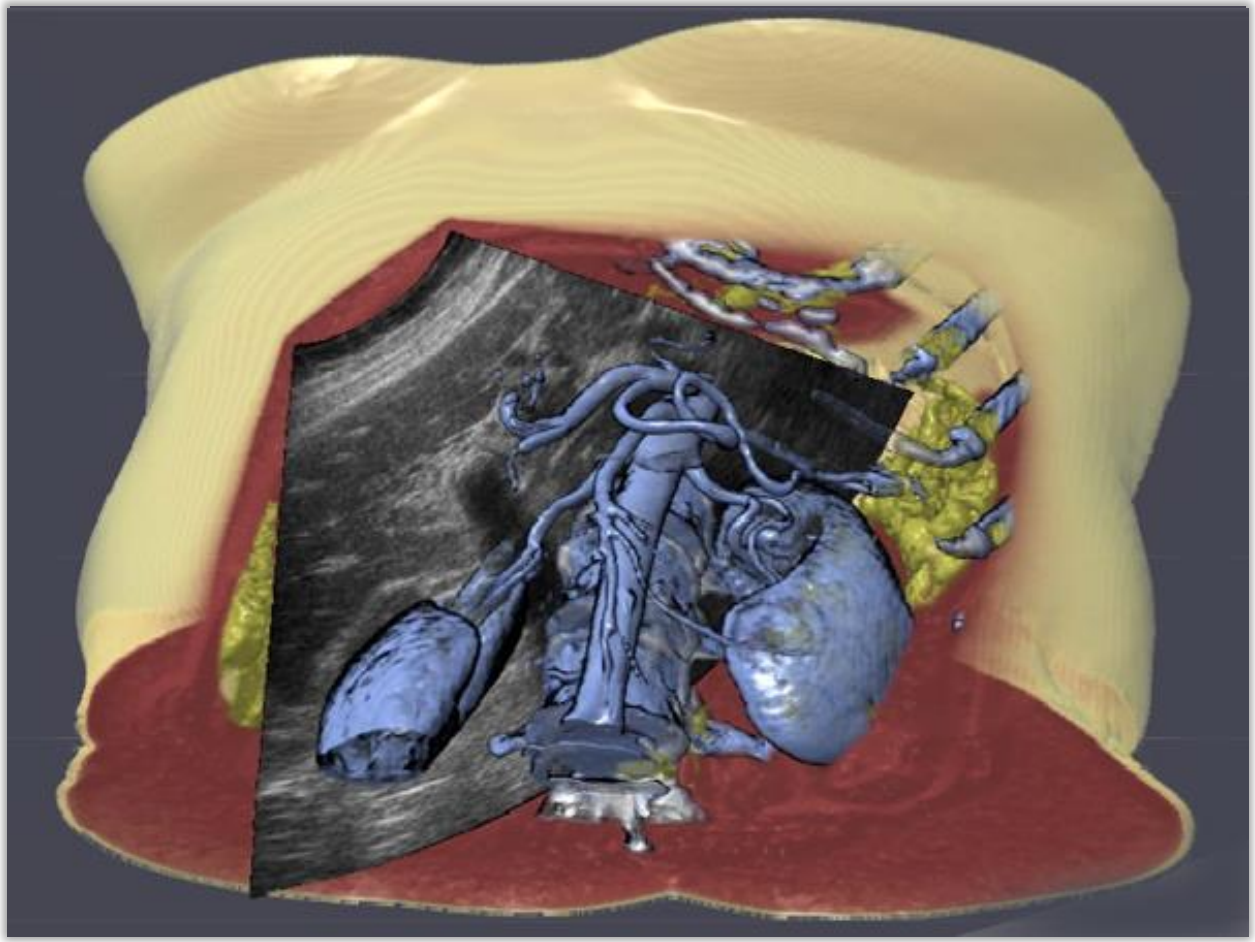
- CT scan with embedded Ultrasound data



Traditional cutaway



Contextual cutaway

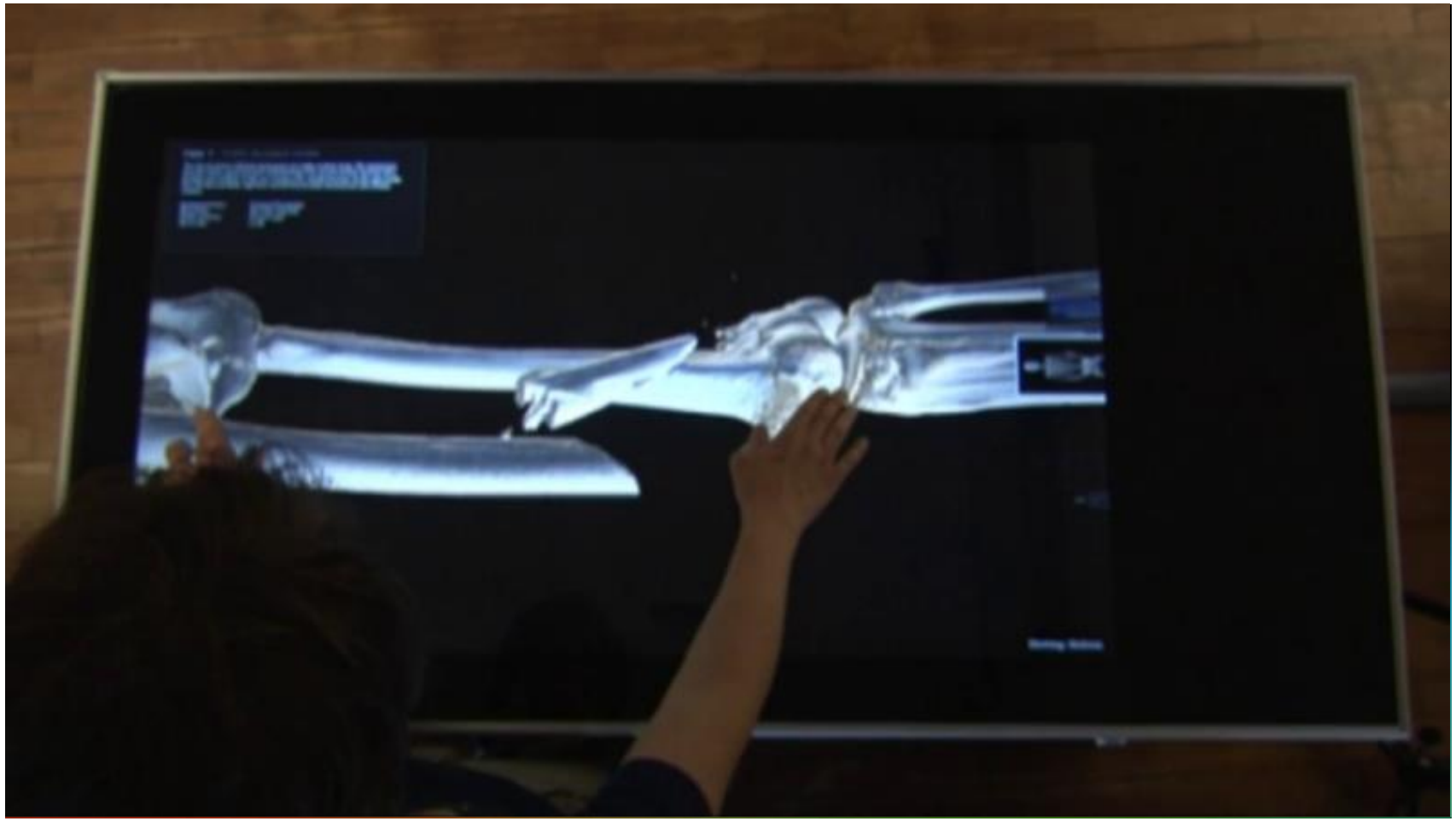


[Burns et al. 2007]

# Visualization – Examples

SIEMENS
Ingenuity for life

- Virtual Autopsy

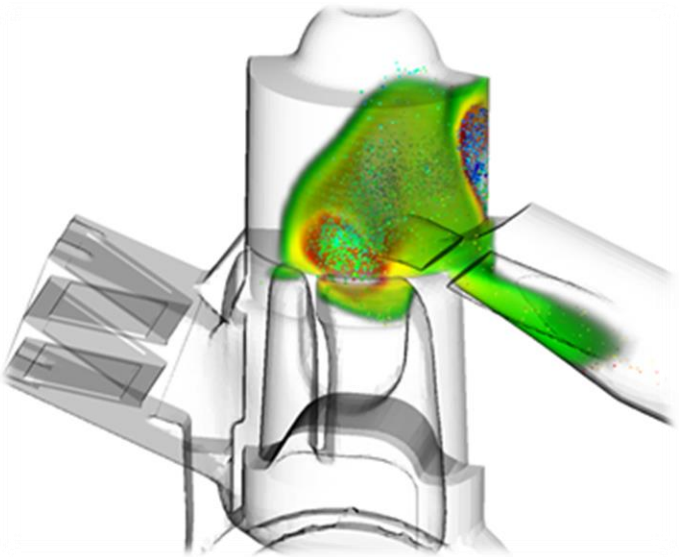


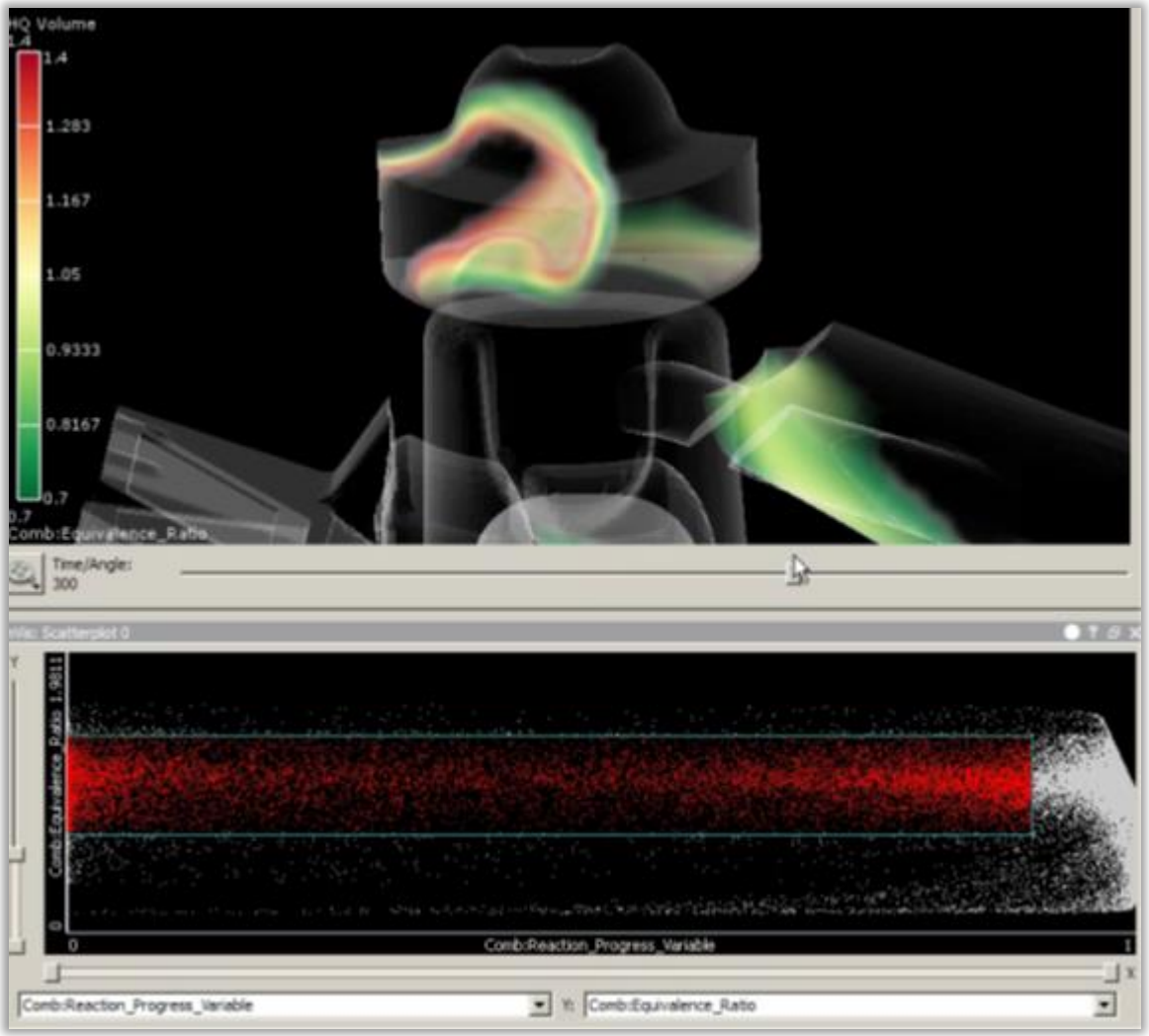
https://vimeo.com/6866296

# Visualization – Examples

SIEMENS
Ingenuity for life

- Combustion process in a two-stroke engine

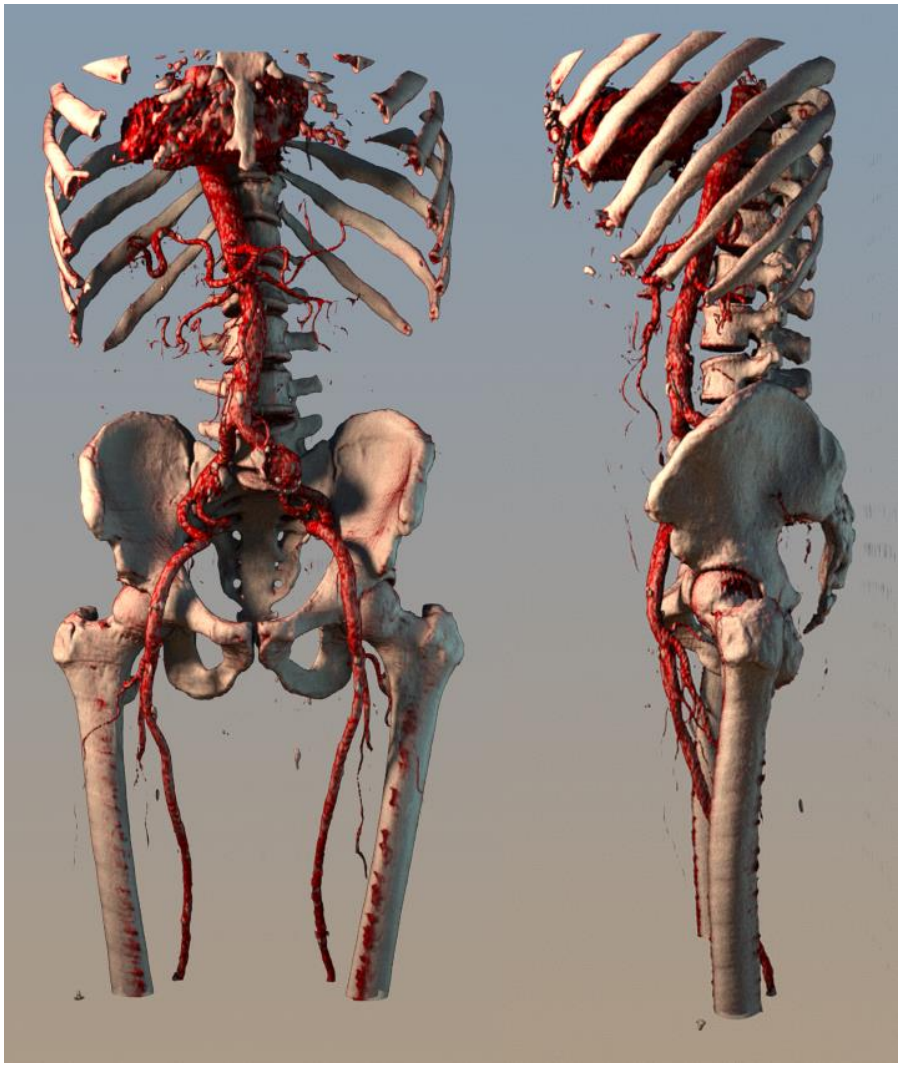




# Visualization – Examples

SIEMENS
Ingenuity for life

- Physically-based lighting



[Kroes et al. 2012]

# Volume visualization

SIEMENS
Ingenuity for life

- Volume rendering techniques
  - Techniques for 2D scalar fields (e.g. slicing)
  - Indirect volume rendering techniques (e.g. surface fitting)
    - Convert/reduce volume data to intermediate representation (surface representation), which can be rendered with traditional techniques – MC-algorithm
  - Direct volume rendering techniques
    - Consider the data as a semi-transparent gel with physical properties and directly get a 3D representation of it

Direct ve indirect volume rendering arasındaki fark nedır ?

Direct volume rendering, veri setinin doğrudan ekrana yansıtılmasını sağlar. Bu yöntem, veri setinin voxel değerlerine göre bir renk ataması yaparak, görüntüde gösterilen nesnelerin kalınlıklarını ve şekillerini belirler. Bu yöntem, veri setinin tamamını bir anda görüntüler ve katmanlı görüntüler oluşturmaz.

Indirect volume rendering ise, veri setinin önceden hesaplanan görüntülerini kullanır. Bu görüntüler, veri setinin voxel değerlerine göre oluşturulan "örtü"lerdir. Örtüler, veri setinin bir kesiti olarak düşünülebilir ve her bir örtü, veri setinin bir zaman diliminde ne gibi değişiklikler gösterdiğini gösterir. Bu örtüler arasında geçiş yaparak, veri setinin değişimlerini izleyebilirsiniz.

Direct volume rendering yöntemi, veri setinin tamamını bir anda görüntülerken, indirect volume rendering yöntemi ise veri setinin değişimlerini zaman içinde izlemek için kullanılır.

Voxel değerleri nedir ve direct volume renderingde nasıl kullanılır

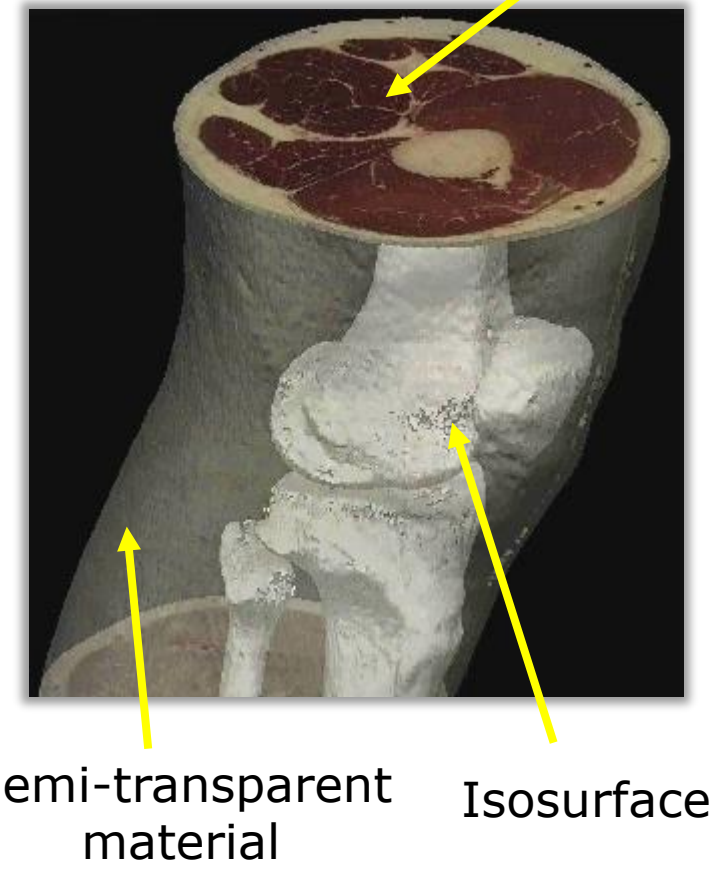
Voxel, veri setlerinde kullanılan 3 boyutlu piksellere verilen isimdir. Her voxel, veri setinin bir noktasını temsil eder ve bu noktanın koordinatları ile birlikte, bir değer içerir. Bu değer, voxelin belirli bir özelliğini temsil eder. Örneğin, bir CT görüntüsünde voxel değerleri, voxelin belirli bir bölgede bulunan doku yoğunluğunu gösterir.

Direct volume rendering yönteminde, voxel değerlerine göre bir renk ataması yapılır. Bu renk ataması, voxelin özelliğini göstermek için kullanılır. Örneğin, bir CT görüntüsünde, düşük yoğunlukta olan dokuların daha açık renklerle, yüksek yoğunlukta olan dokuların daha koyu renklerle gösterilmesi mümkündür. Bu renk ataması sayesinde, veri setindeki nesnelerin kalınlıkları ve şekilleri belirlenir ve görüntüde görülebilir hale gelir.

# Volume visualization

SIEMENS
Ingenuity for life

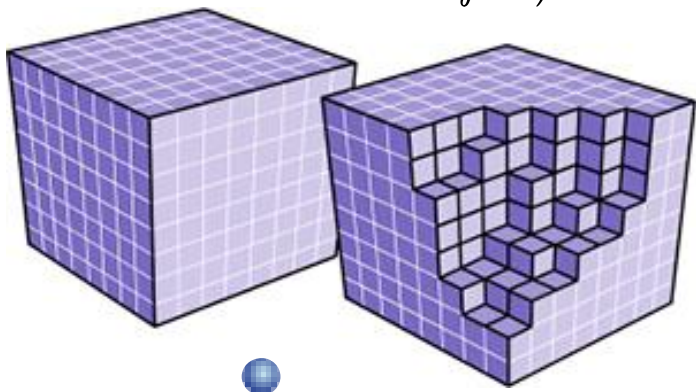
- **Slicing**
  Display volume data, mapped to colors, on a slice plane
- **Iso-surfacing**
  Generate opaque/semi-opaque surfaces (e.g., via the MC-Algorithm)
- **Direct Volume Rendering**
  Volume material attenuates reflected or emitted light

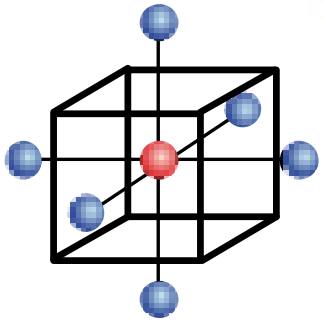


# Volume visualization

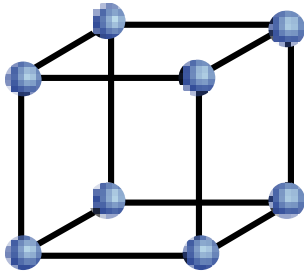
SIEMENS
Ingenuity for life

- Data values are initially given at vertices of a 3D grid
  - These are called voxels (volume elements)
  - Voxel = point sample in 3D





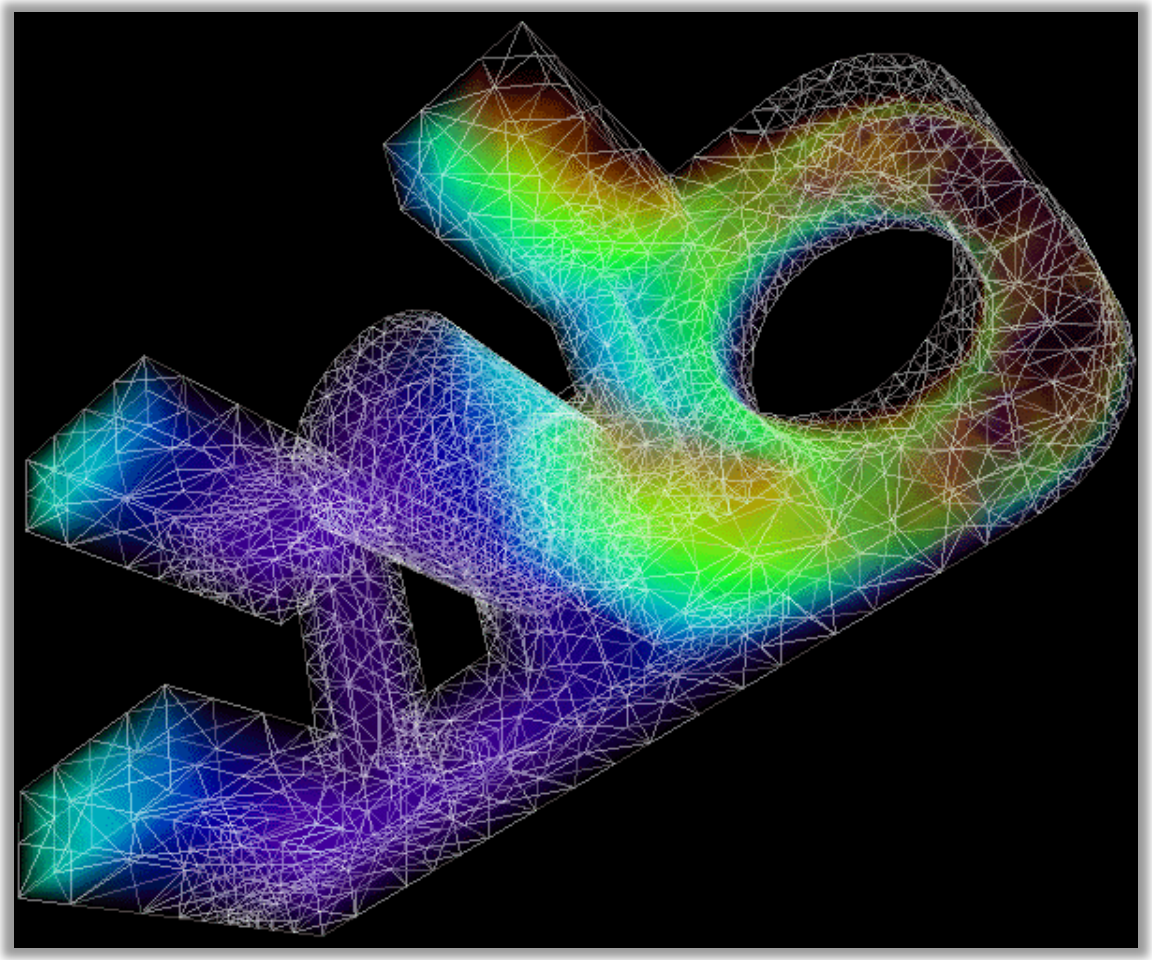
- "Adjacent" grid vertices make up a cell
  - Use interpolation for data reconstruction in a cell
  - Corners: 8 voxel



# Volume visualization

SIEMENS
Ingenuity for life

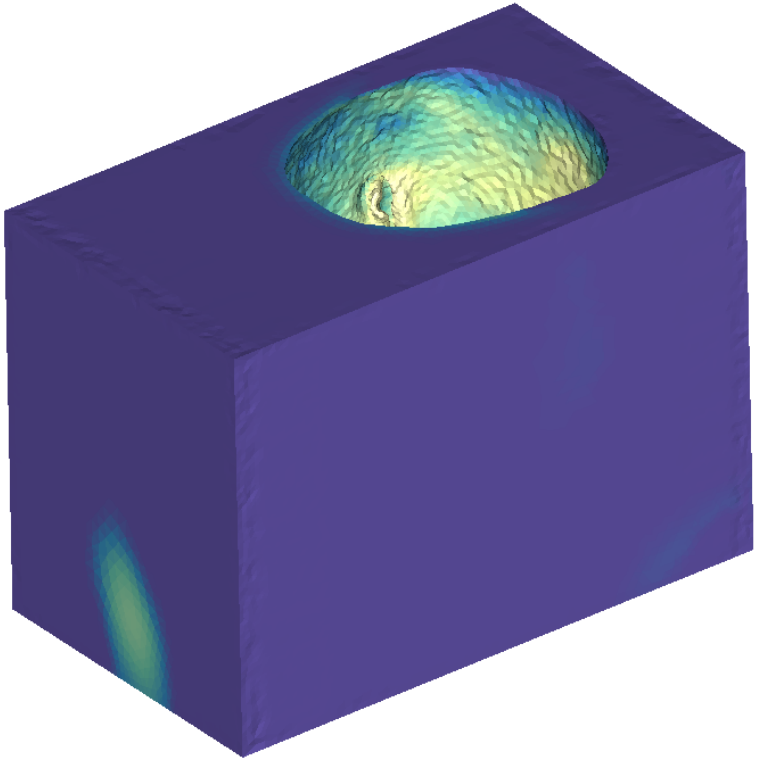
- A volumetric object has…
  - a shape (given by the geometry of the 3D grid)
  - an appearance (given by the data values)

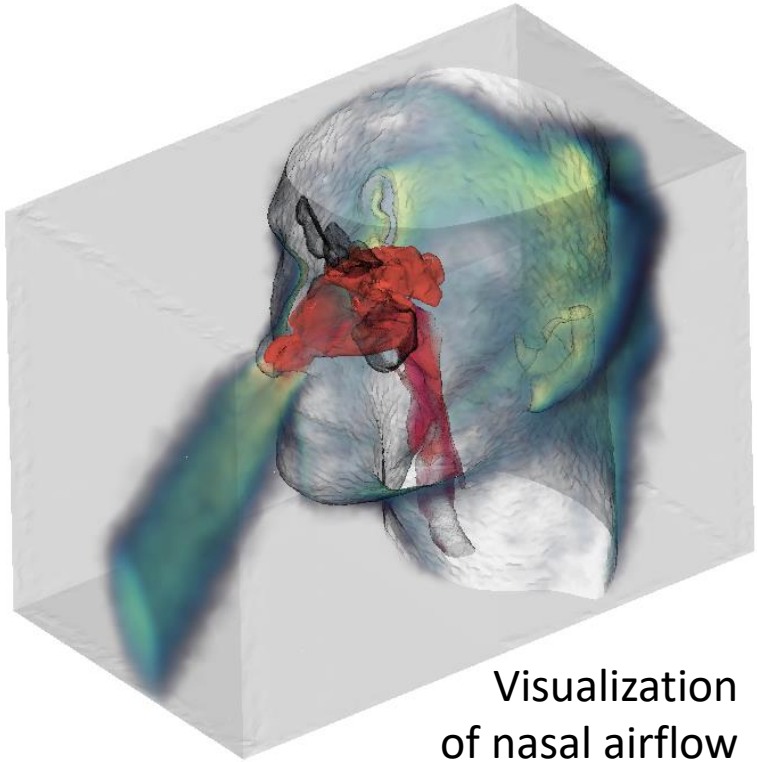


# Volume visualization

SIEMENS
Ingenuity for life

- Some characteristics of volume data
  - Essential information in the interior
  - Often cannot be described by a surface in general (e.g., fire, clouds, gaseous phenomena)



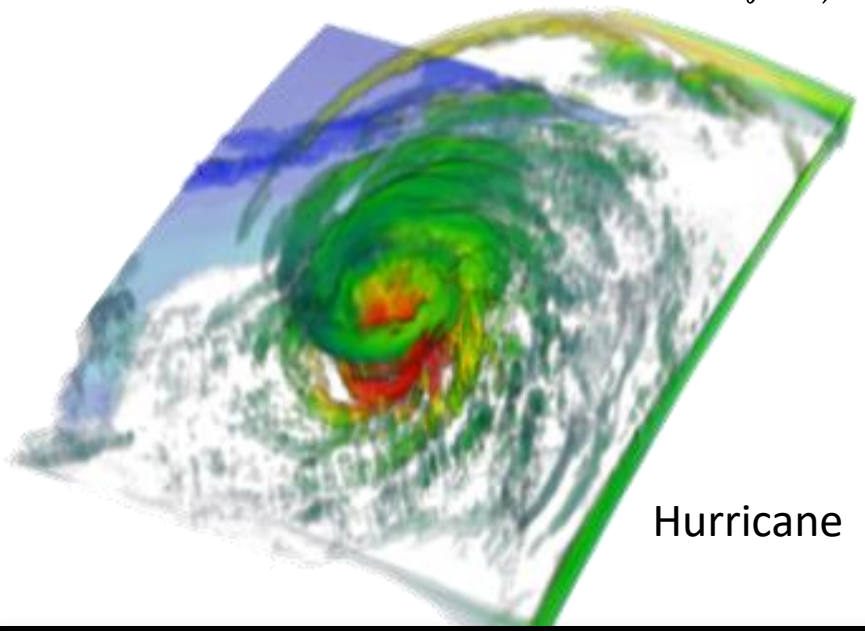


Visualization of nasal airflow

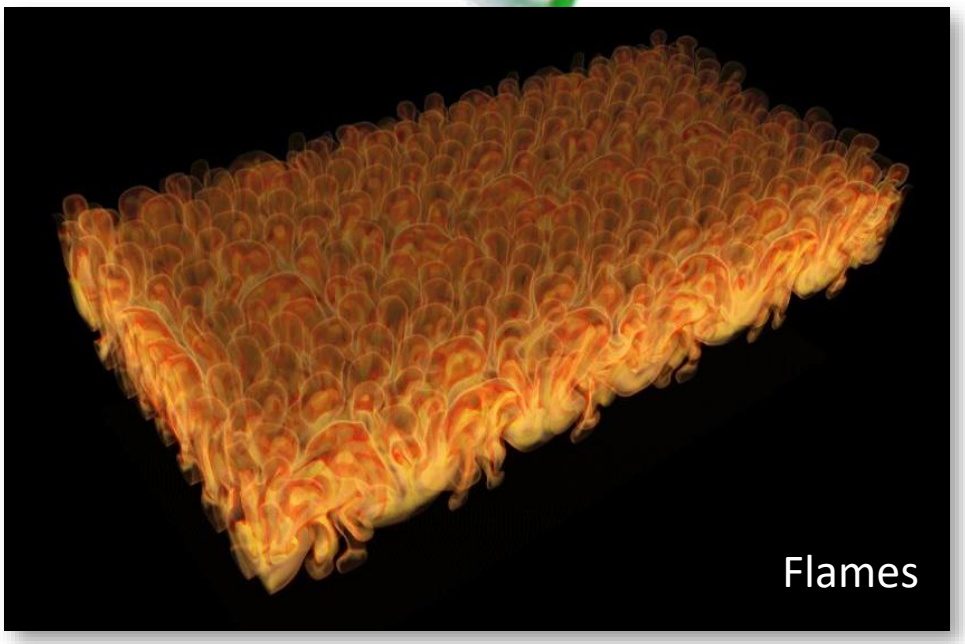
# Volume visualization

SIEMENS
Ingenuity for life

Clouds



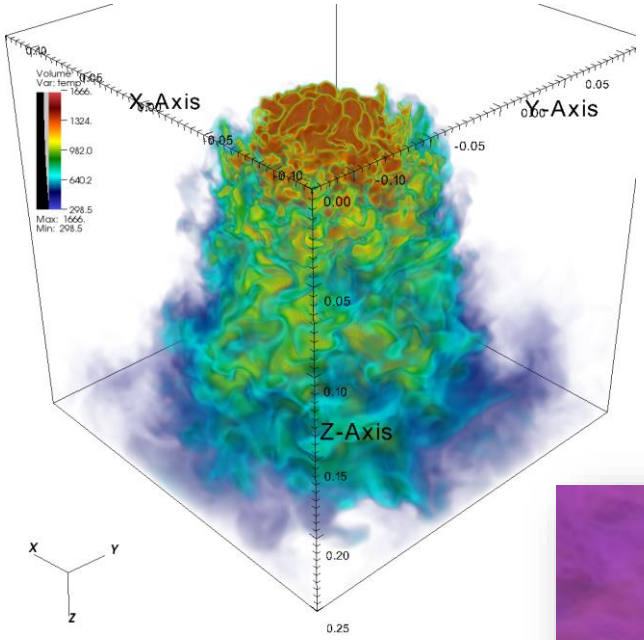
Hurricane



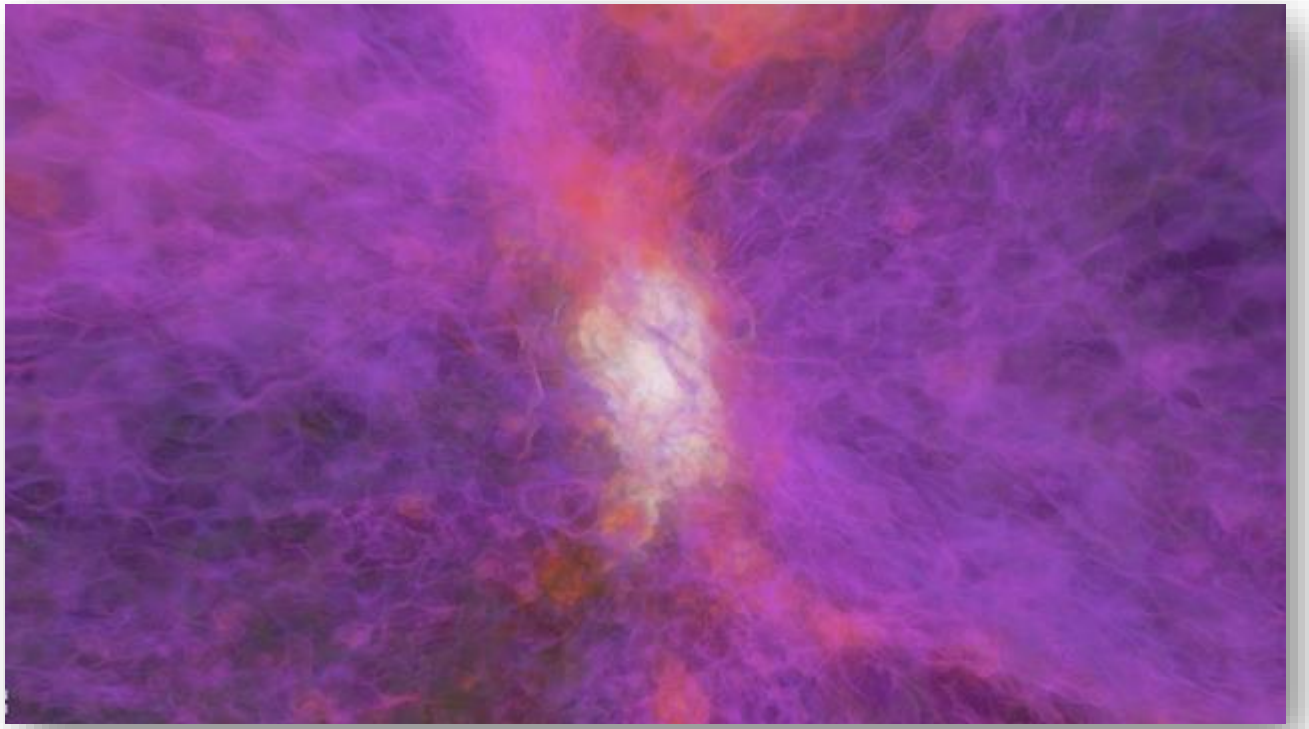
Flames

# Volume visualization

SIEMENS
Ingenuity for life



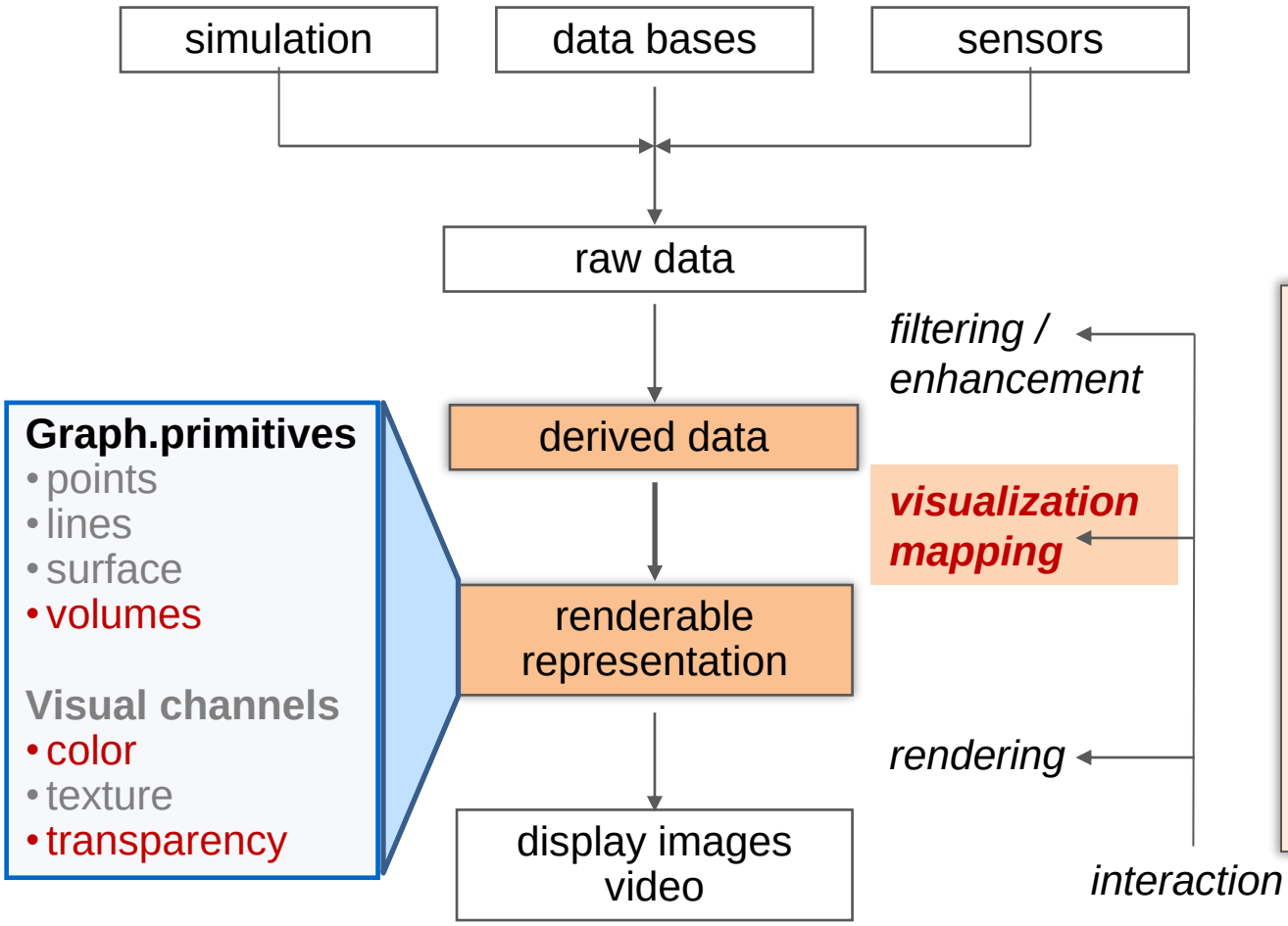

Hydrogen flame



Millennium Simulation: Distribution of matter in the Universe

# Volume Visualization

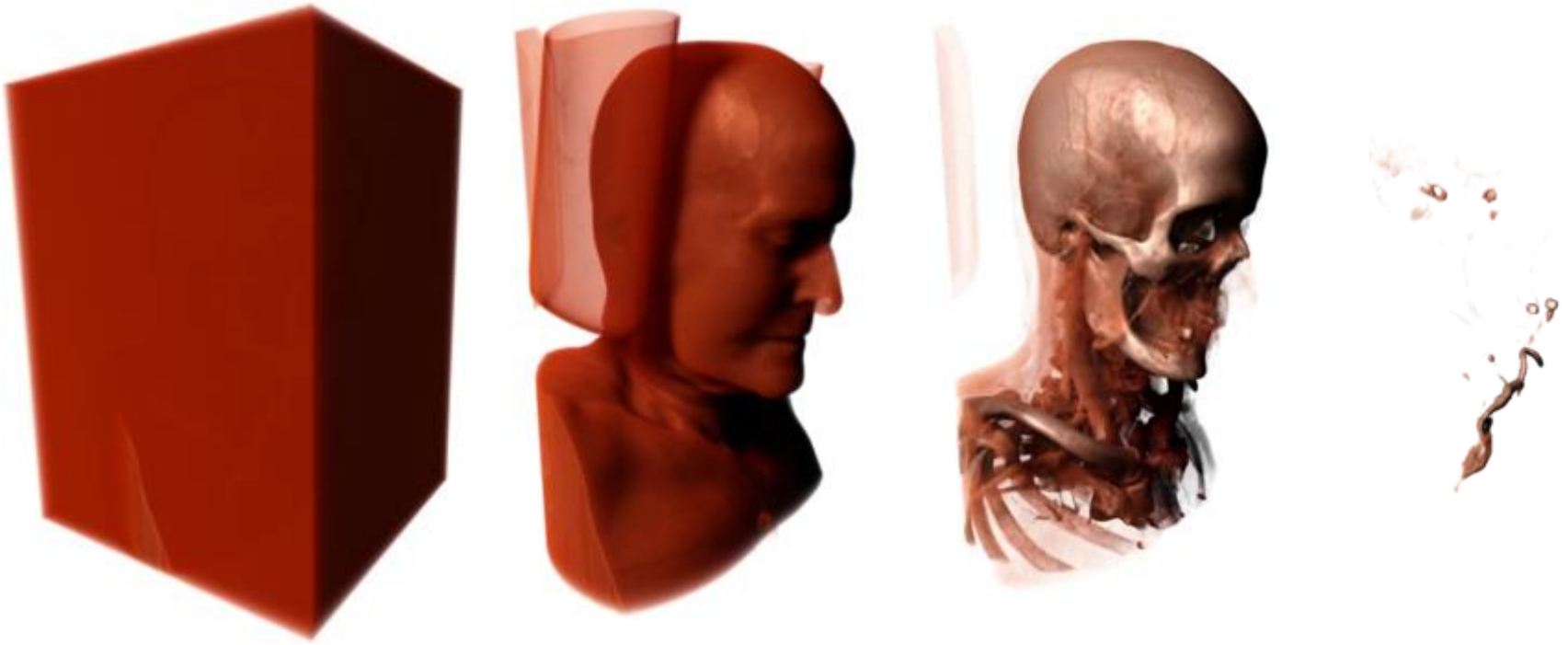
SIEMENS
Ingenuity for life



# Volume visualization

SIEMENS
Ingenuity for life

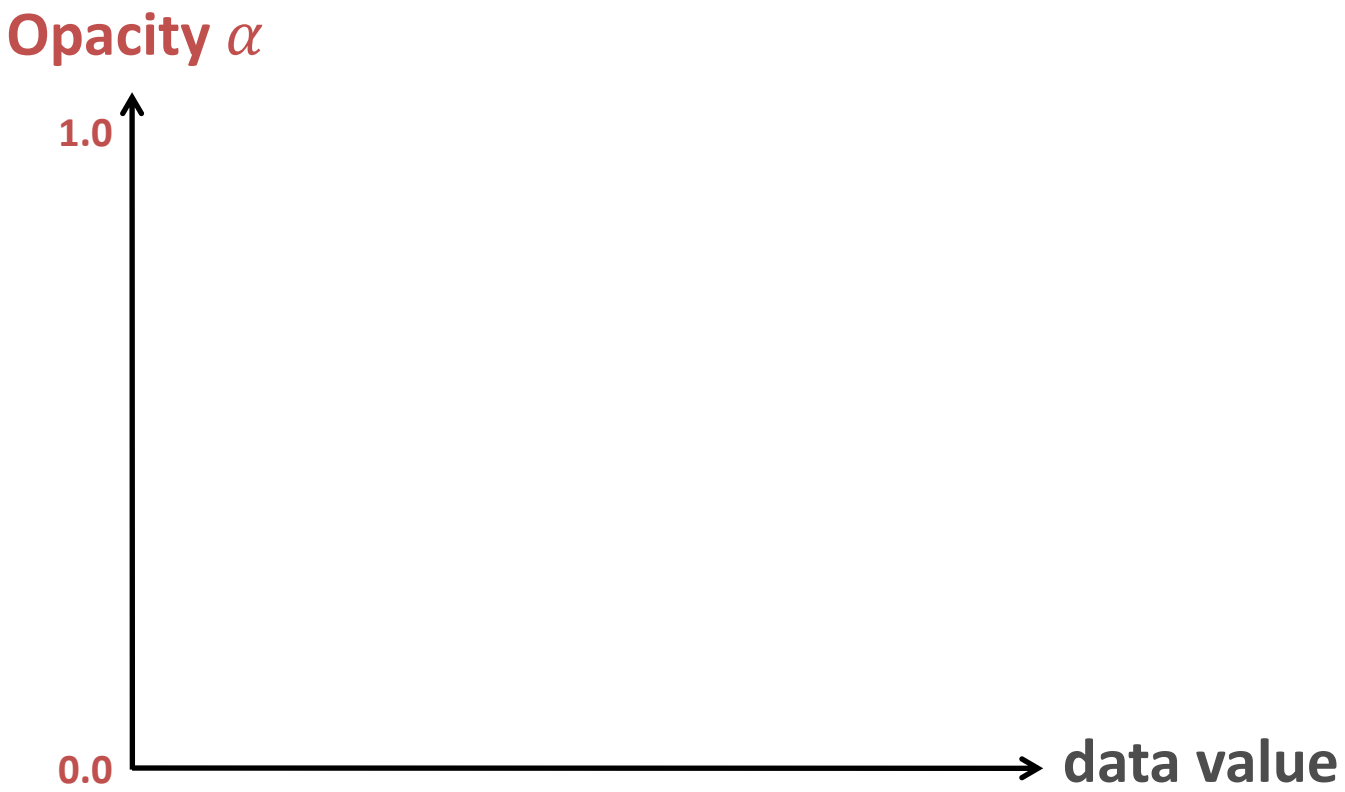
- Before data is visualized, it has to be classified, i.e., data values are mapped to visual properties like color and opacity
- Mapping is performed via a transfer function



# Volume visualization

SIEMENS
Ingenuity for life

- Transfer function
  - Associate distinct materials (value ranges) to distinct properties (color & opacity)



# Volume visualization

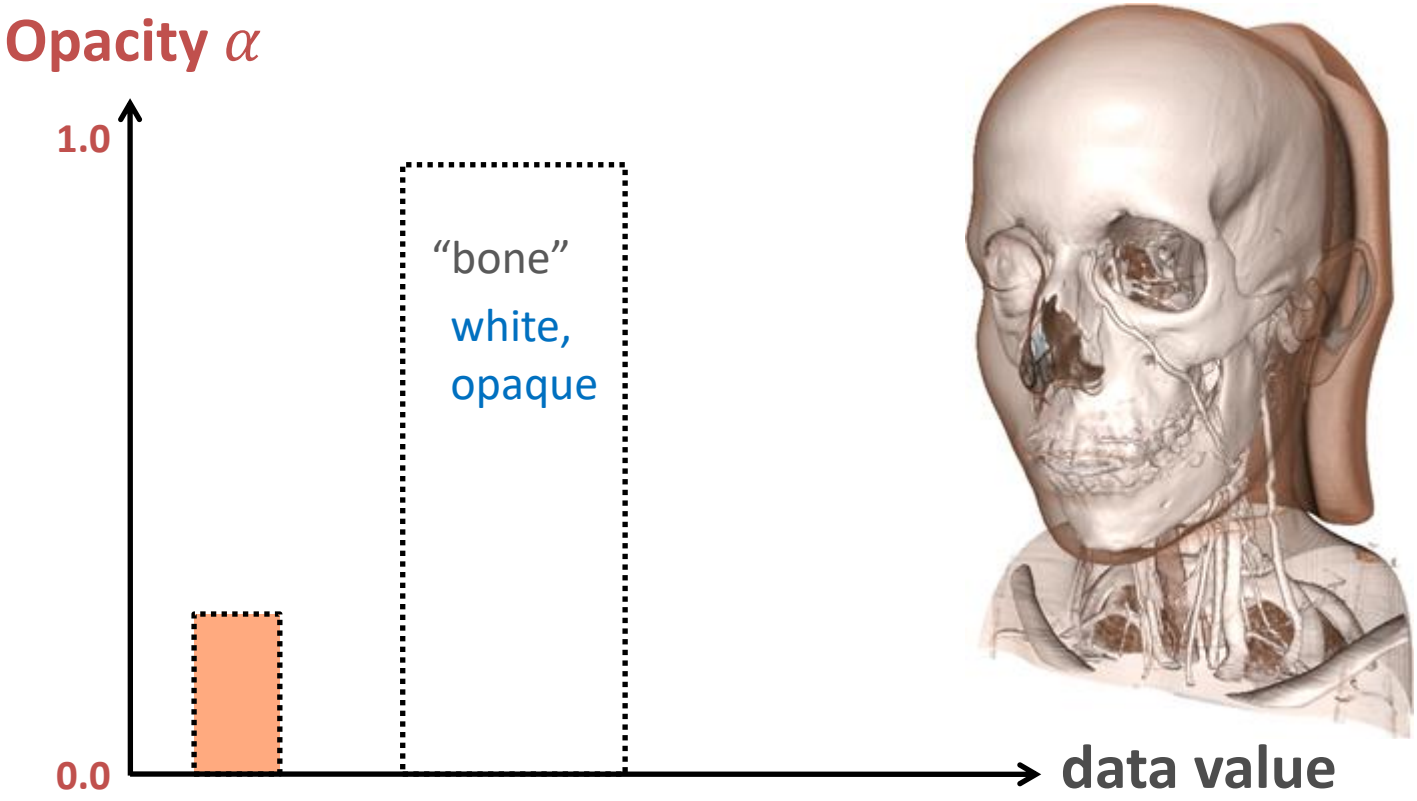
SIEMENS
Ingenuity for life

- Transfer function
  - Data value → color: $f(\boldsymbol{x}) \rightarrow C(\boldsymbol{x})$
  - Data value → opacity: $f(\boldsymbol{x}) \rightarrow \alpha(\boldsymbol{x})$

# Volume visualization

SIEMENS
Ingenuity for life

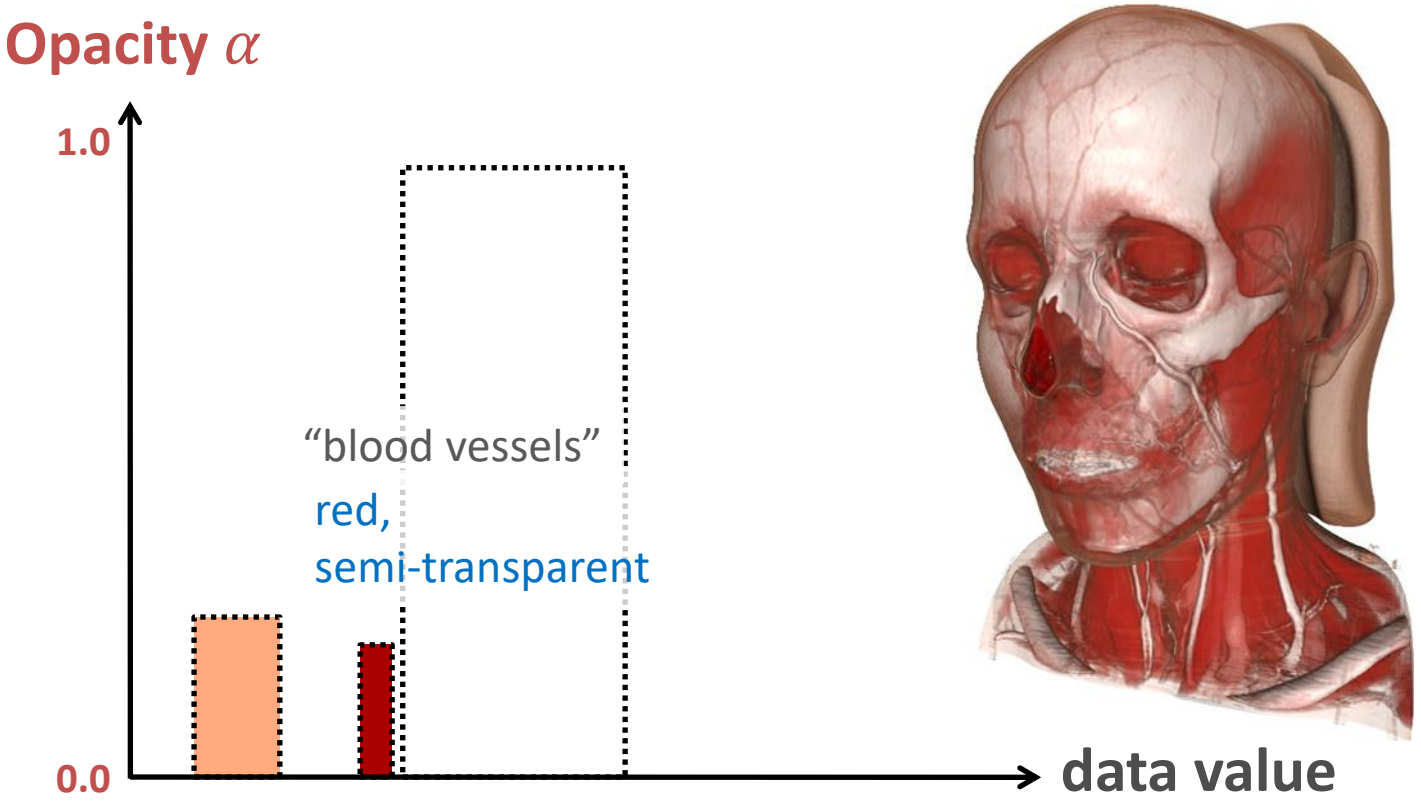
- Transfer function
  - Data value → color: $f(\boldsymbol{x}) \rightarrow C(\boldsymbol{x})$
  - Data value → opacity: $f(\boldsymbol{x}) \rightarrow \alpha(\boldsymbol{x})$



# Volume visualization

SIEMENS
Ingenuity for life

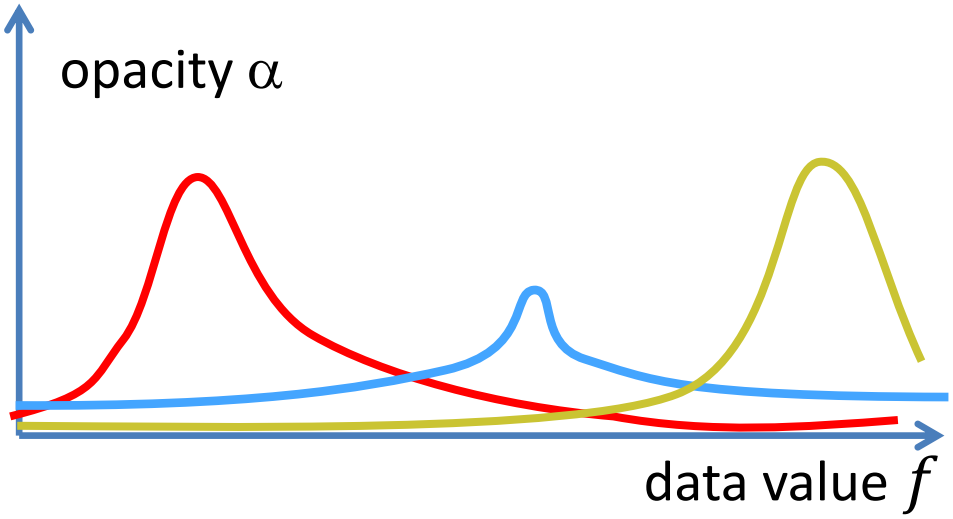
- Transfer function
  - Data value → color: $f(\boldsymbol{x}) \rightarrow C(\boldsymbol{x})$
  - Data value → opacity: $f(\boldsymbol{x}) \rightarrow \alpha(\boldsymbol{x})$

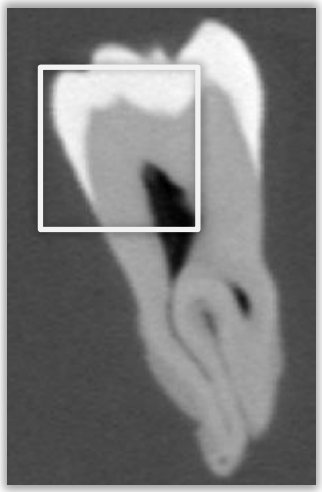


# Volume visualization

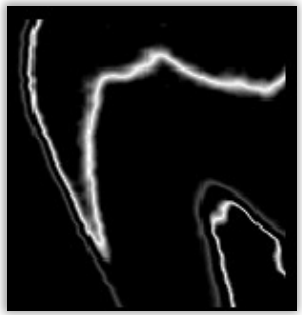
SIEMENS
Ingenuity for life

- Transfer function design

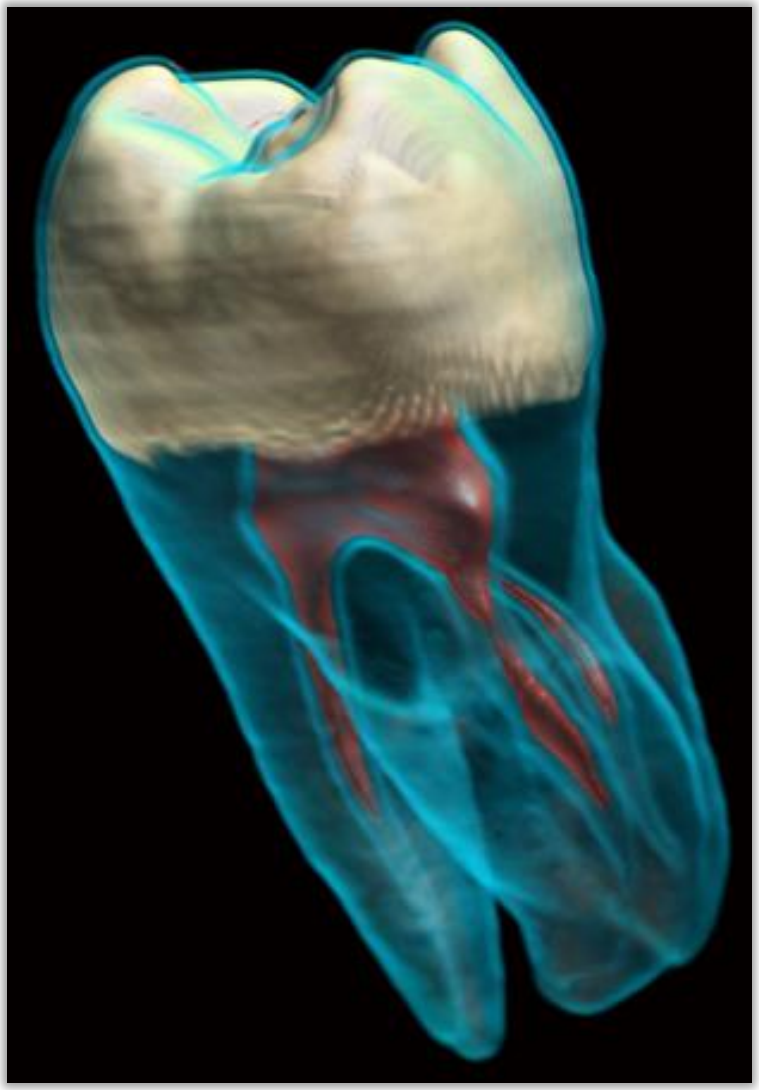


color



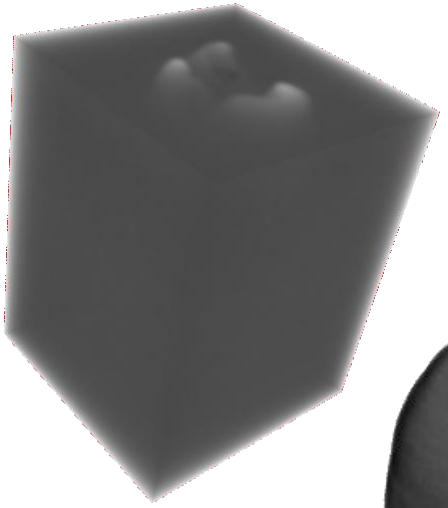
opacity

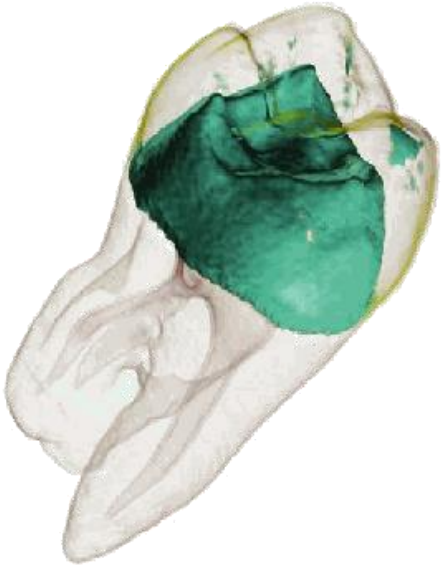


# Volume visualization

SIEMENS
Ingenuity for life

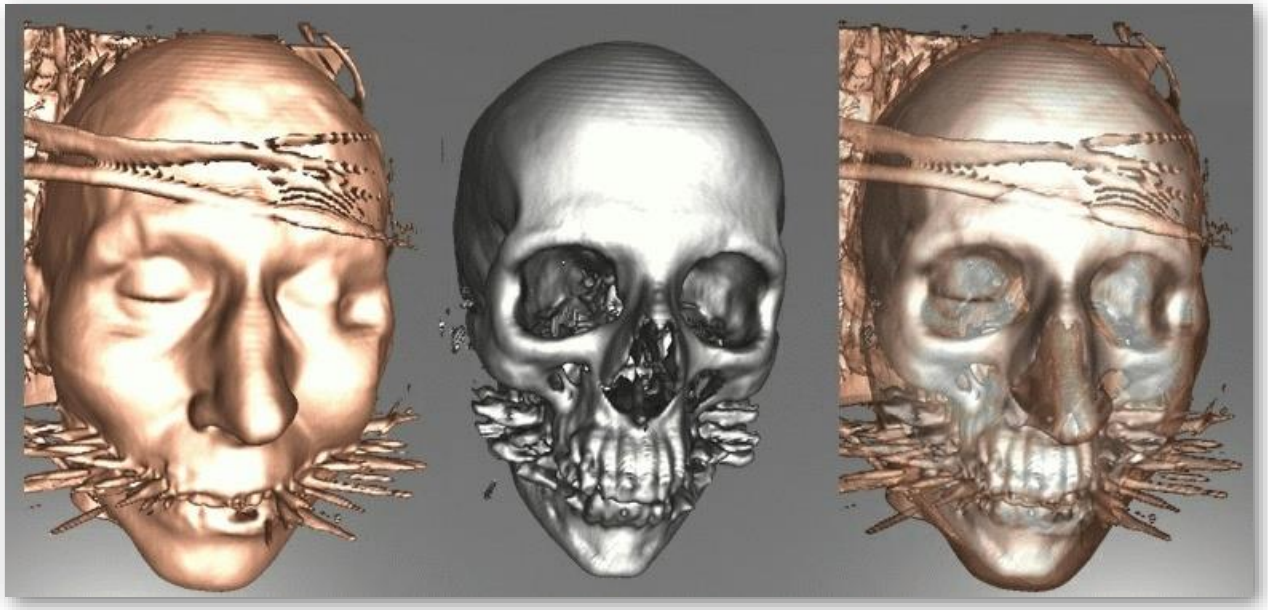
- Examples of different transfer functions for the same dataset

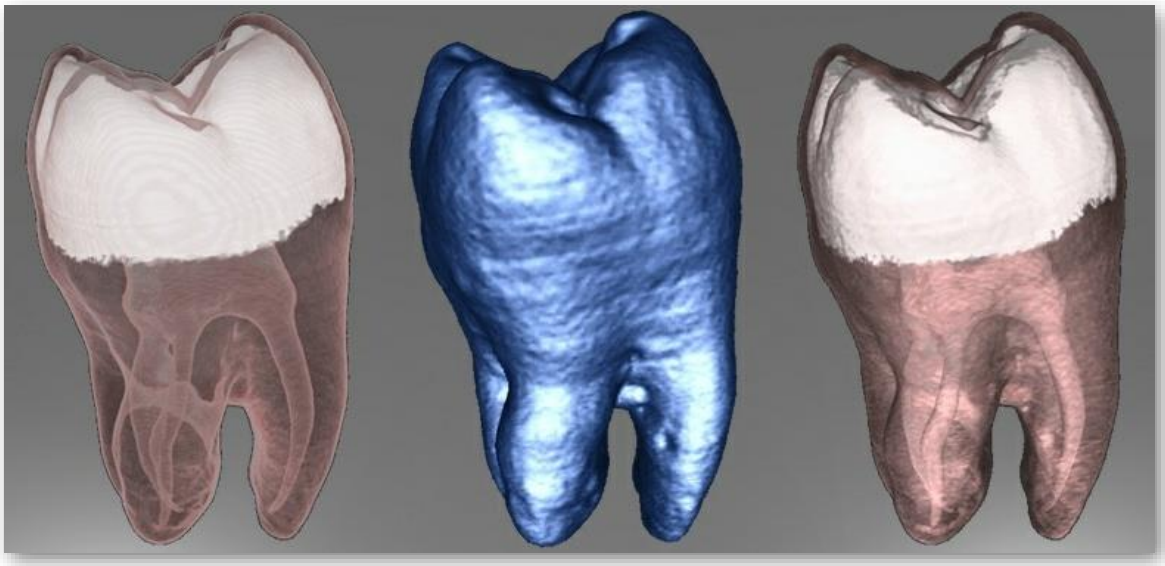




# Volume visualization

SIEMENS
Ingenuity for life



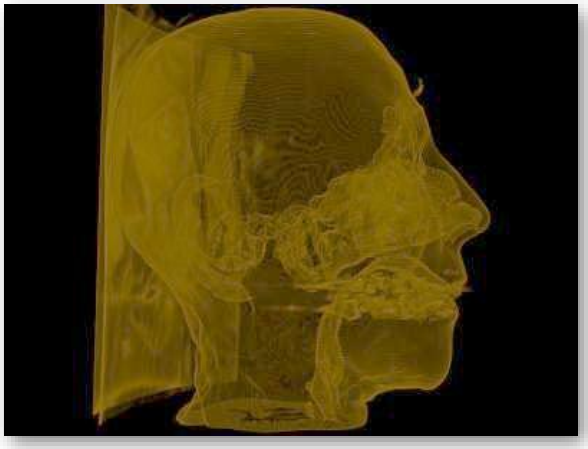


[Engel et al. 06, Krüger & Westermann 03]

# Volume visualization

SIEMENS
Ingenuity for life

- Note that color is a material property
- Note that opacity $\alpha$ is also such a property and can be assigned as well!
- Color is then a quadrupel (RGB$\alpha$)

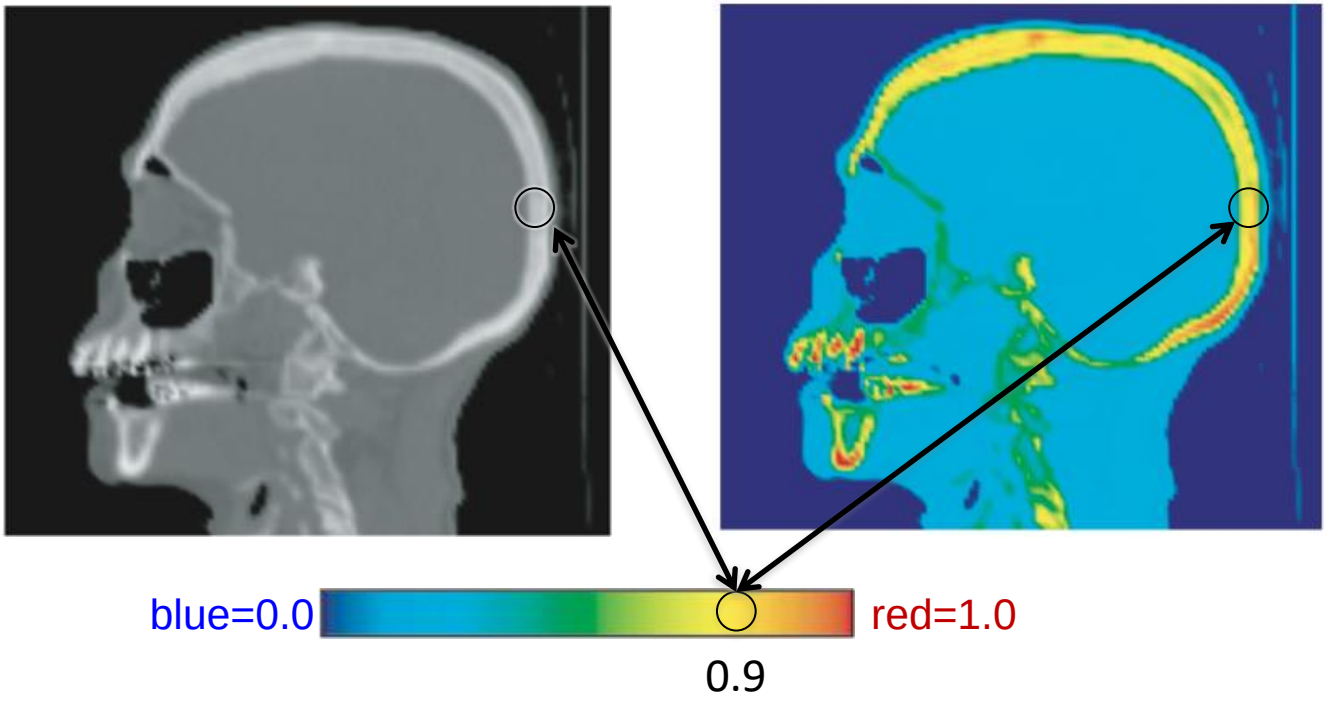




# Volume visualization

SIEMENS
Ingenuity for life

- Color mapping for scalar data
  - Assign a different color to each scalar value
  - Assignment via a so-called **transfer function** *T*

    *T : scalar value → color + α-value*

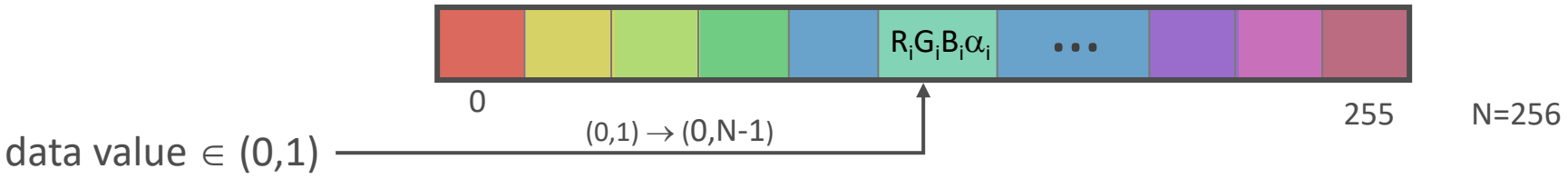


# Volume visualization

SIEMENS
Ingenuity for life

- Color mapping for scalar data
  - Assign a different color to each scalar value
  - Assignment via a so-called **transfer function** *T*

    *T : scalar value → color + α-value*
  - Realization
    - RGBA-color values are stored in a color table
    - For a data value, the color at the corresponding entry in the table is used



# Direct volume rendering

SIEMENS
Ingenuity for life

- Get a 3D representation of the volume data taking into account emission and absorption

# Direct volume rendering

SIEMENS
Ingenuity for life

- Direct volume rendering (DVR) considers the physics of light transport in a dense medium
  - Optical properties are mapped to each voxel (emission = color, absorption = opacity)
  - The light reaching the viewer is simulated by ray casting

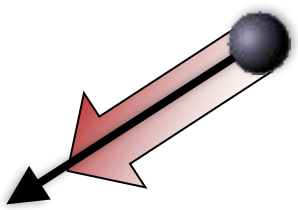
No need to extract intermediate explicit geometry (iso-surface)

# Direct volume rendering

SIEMENS
Ingenuity for life

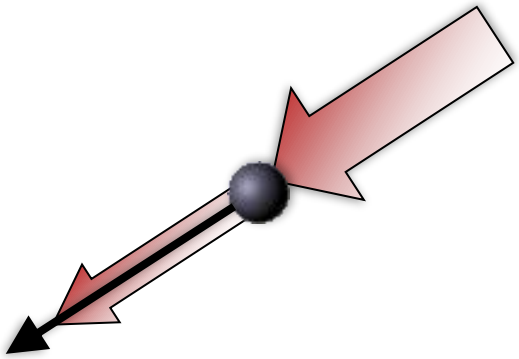
- Direct volume rendering
  - The collection of light along straight rays is based on an emission/absorption model
  - Assumption: Volume consists of small particles
  - Each voxel emits light of the color which is assigned to it, and absorbs light according to it's opacity

Emission



Energy increase

Absorption

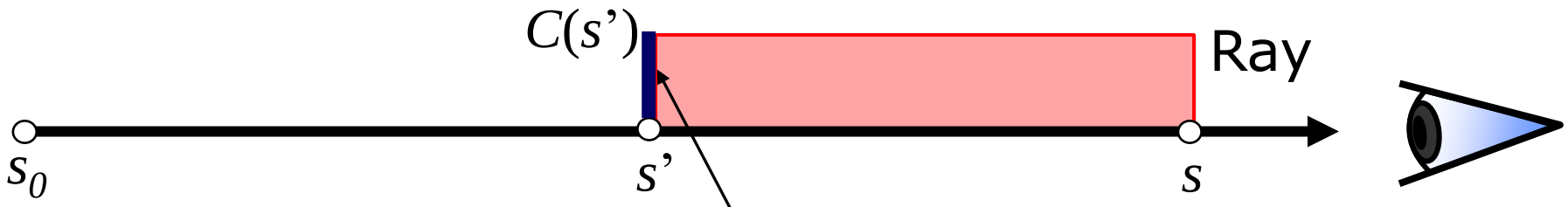


Energy decrease

# Light emission and attenuation

SIEMENS
Ingenuity for life

- The volume rendering integral




Active color emission at $s$'

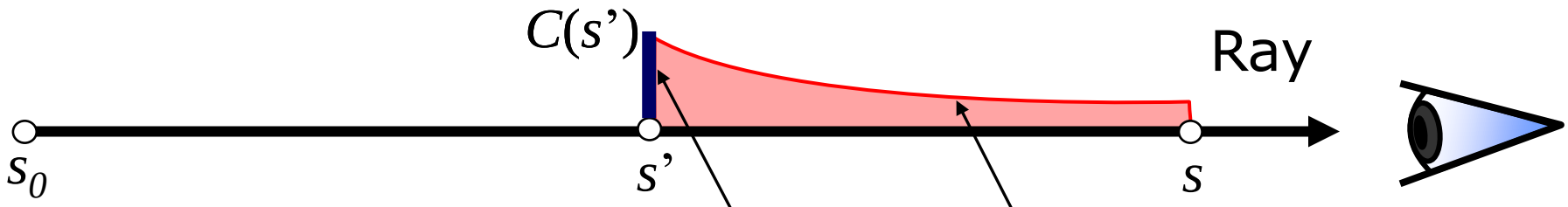
$$C(s) = C(s')$$

Without absorption the color emitted at point $s$' would reach point $s$

# Light emission and attenuation

SIEMENS
Ingenuity for life

- The volume rendering integral




Active color emission at $s'$

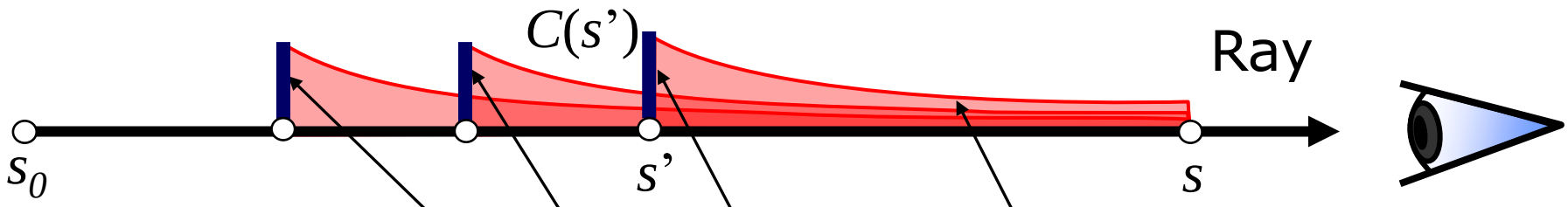
Absorption along the distance $s'$- $s$ caused by accumulated opacity

$$C(s) = C(s') \cdot e^{-\int_{s'}^{s} \alpha(t)\, dt}$$

# Light emission and attenuation

SIEMENS
Ingenuity for life

- The volume rendering integral




For all sample points $s'$

Active color emission at $s'$

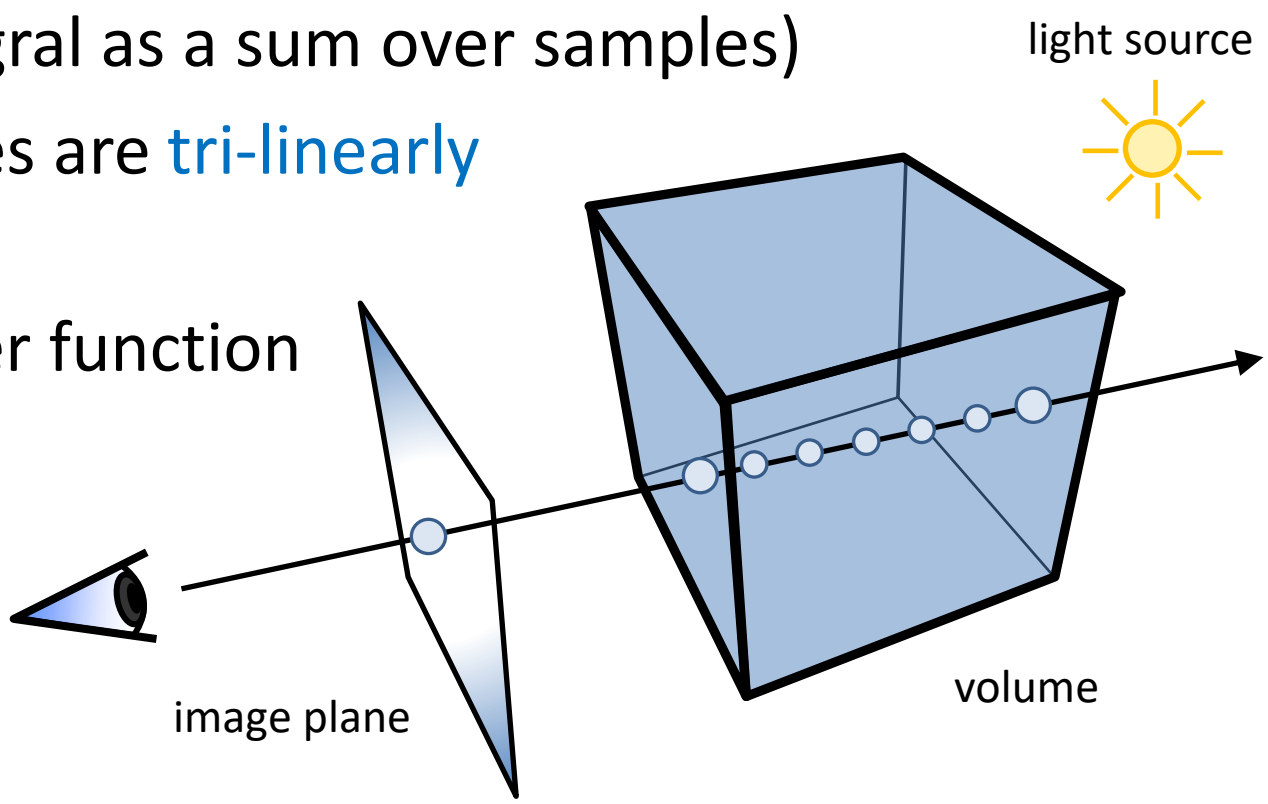
Absorption along the distance $s'$- $s$ caused by accumulated opacity

$$C(s) = \int_{s_0}^{s} C(s') \cdot e^{-\int_{s'}^{s} \alpha(t)\, dt}\, ds'$$

# Direct volume rendering

SIEMENS
Ingenuity for life

- **Ray-casting:** Numerical approximation of the volume rendering integral
  - A ray is cast into volume for each output pixel
  - Volume is resampled at equidistant intervals along the ray (integral as a sum over samples)
  - Sample values are tri-linearly interpolated
  - Apply transfer function



# Direct volume rendering

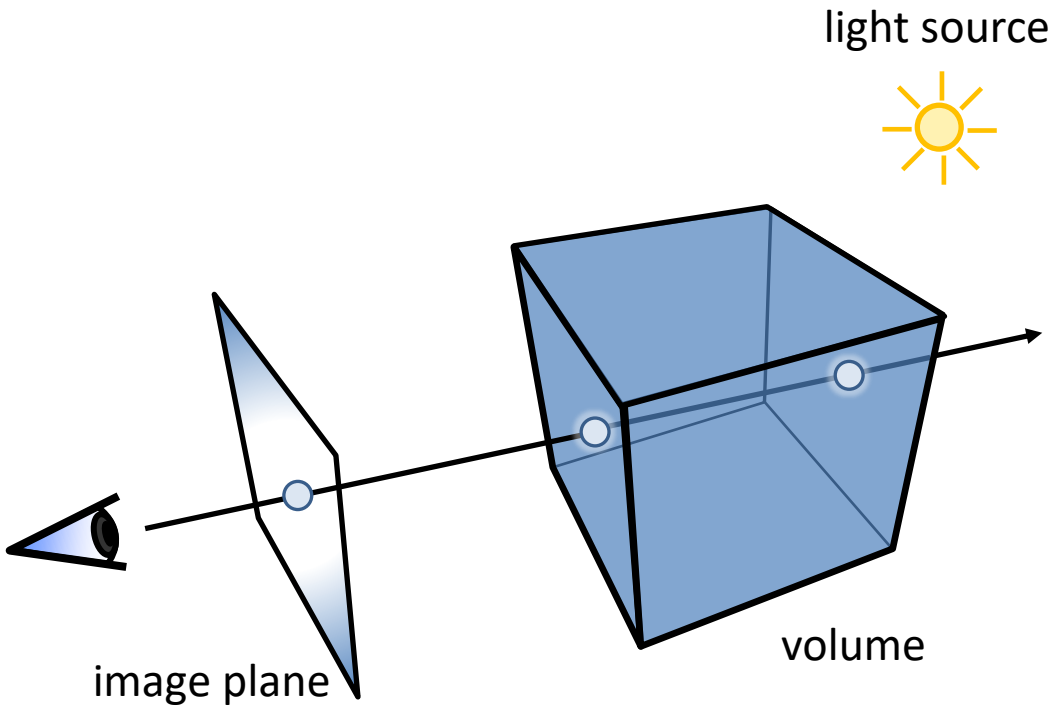
SIEMENS
Ingenuity for life

- Ray-casting method
  - Defines a virtual image plane where viewer is looking through
  - Cast a ray through every pixel on the screen

**for each** pixel on the image plane

    compute entry- and exit-point in volume

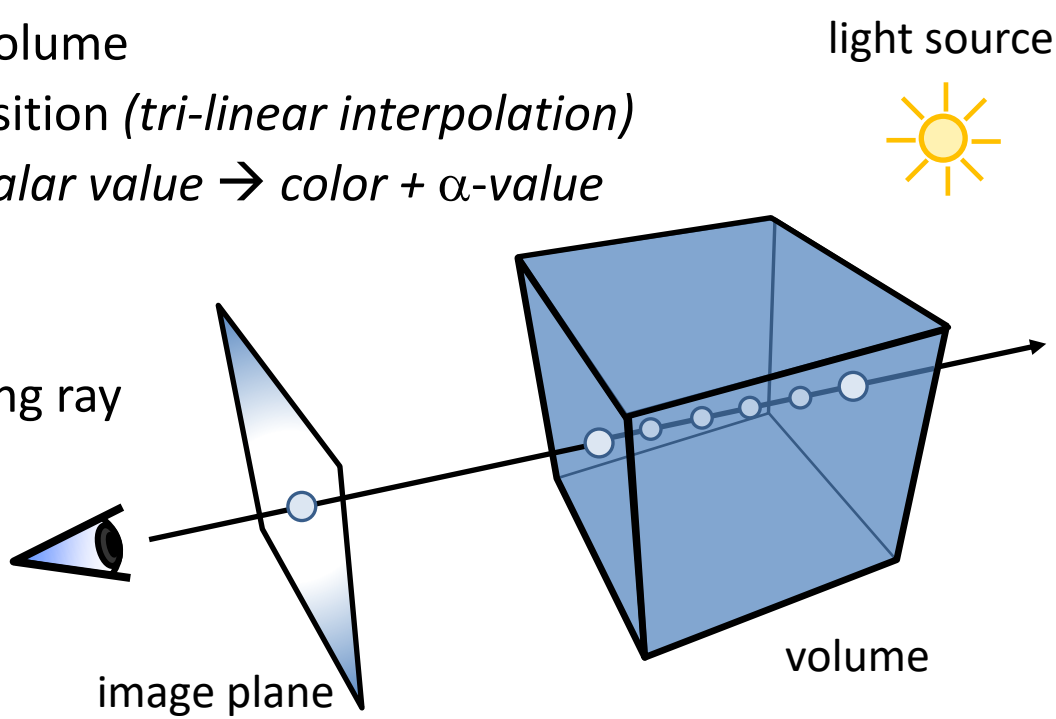
**end for**



# Direct volume rendering

- Ray-casting method
  - Defines a virtual image plane where viewer is looking through
  - Cast a ray through every pixel on the screen

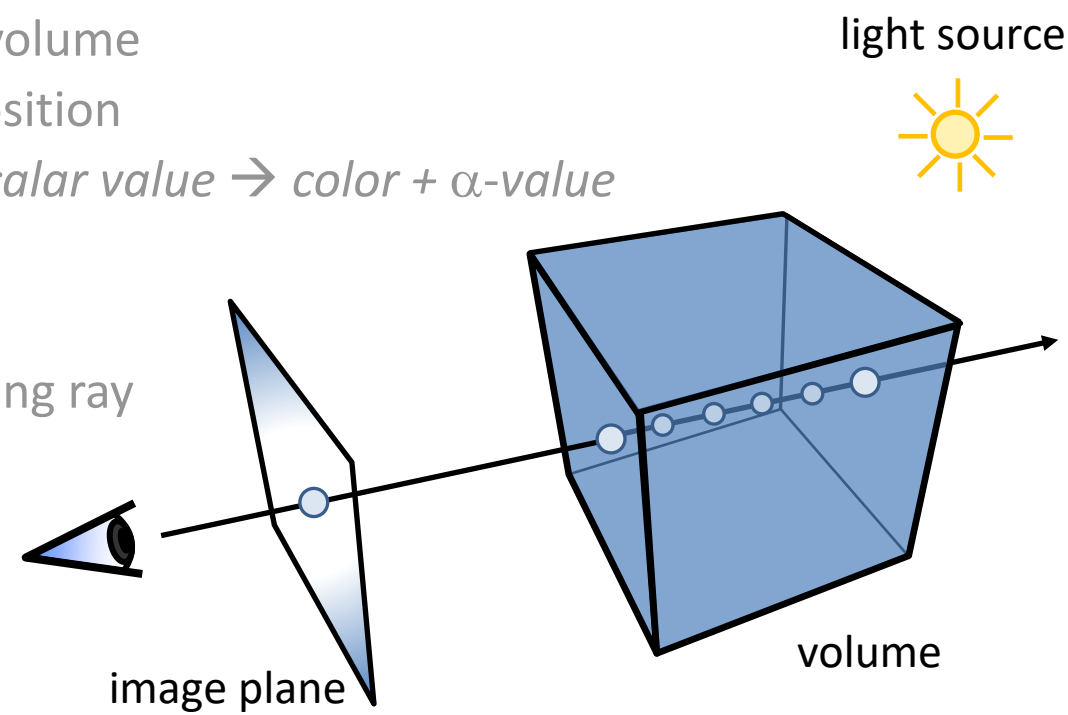
**for each** pixel on the image plane
    compute entry- and exit-point in volume
    **while** current position inside volume
        read density at current position *(tri-linear interpolation)*
        apply transfer function: *scalar value* → *color* + $\alpha$-*value*
        compute new position along ray
    **end while**
**end for**



# Direct volume rendering

- Ray-casting method
  - Defines a virtual image plane where viewer is looking through
  - Cast a ray through every pixel on the screen

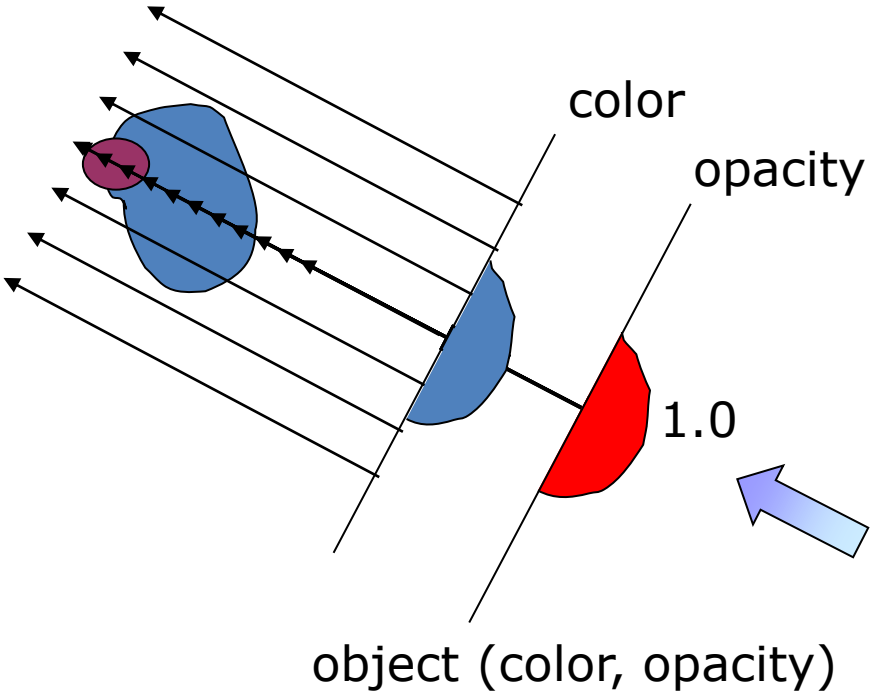
```
for each pixel on the image plane
    compute entry- and exit-point in volume
    while current position inside volume
        read density at current position
        apply transfer function: scalar value → color + α-value
        compute shading
        apply compositing
        compute new position along ray
    end while
    set pixel color in image plane
end for
```



# Ray casting

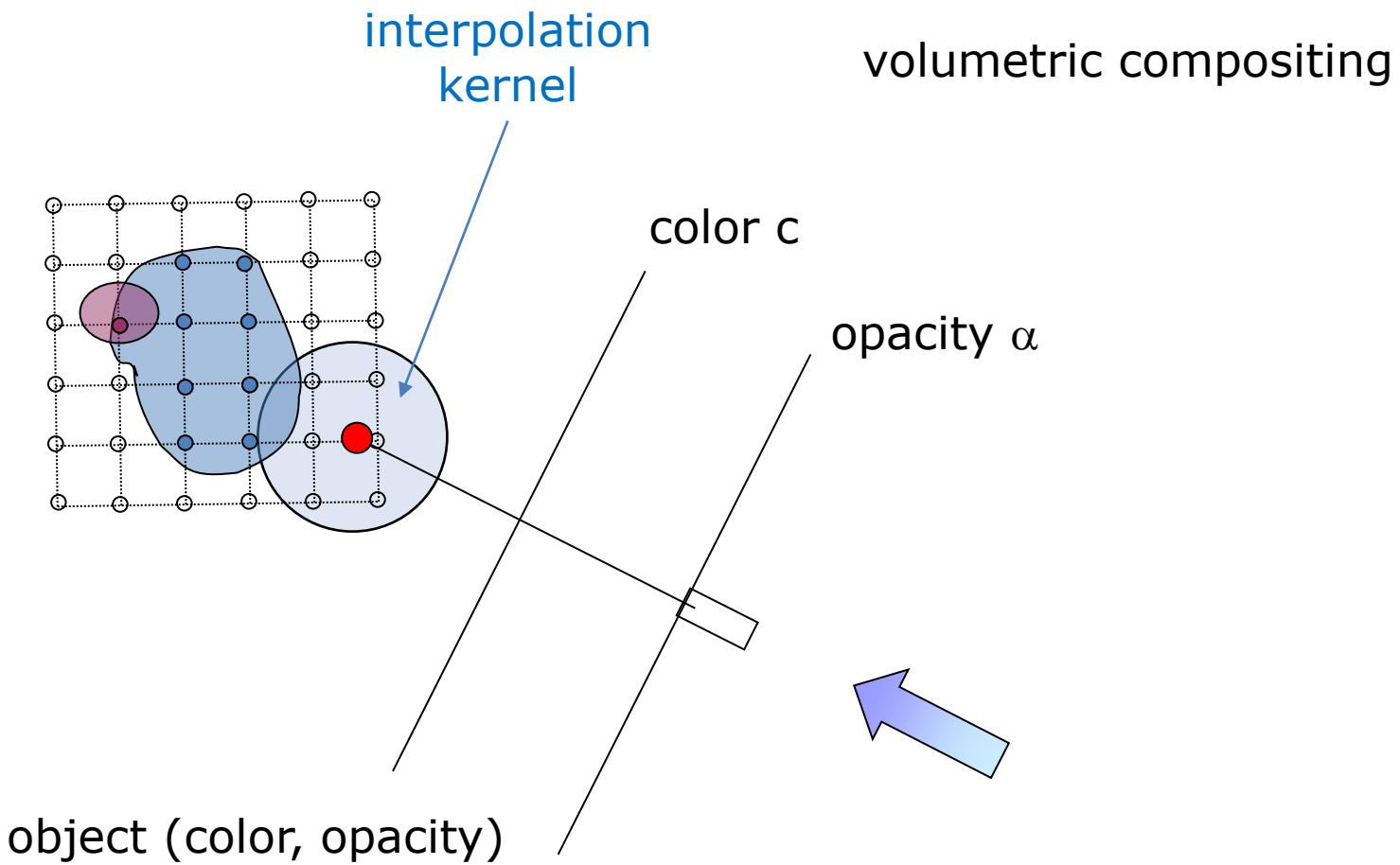
SIEMENS
Ingenuity for life

- Volumetric ray integration:
  - Compositing: accumulation of color & opacity along rays



# Ray casting

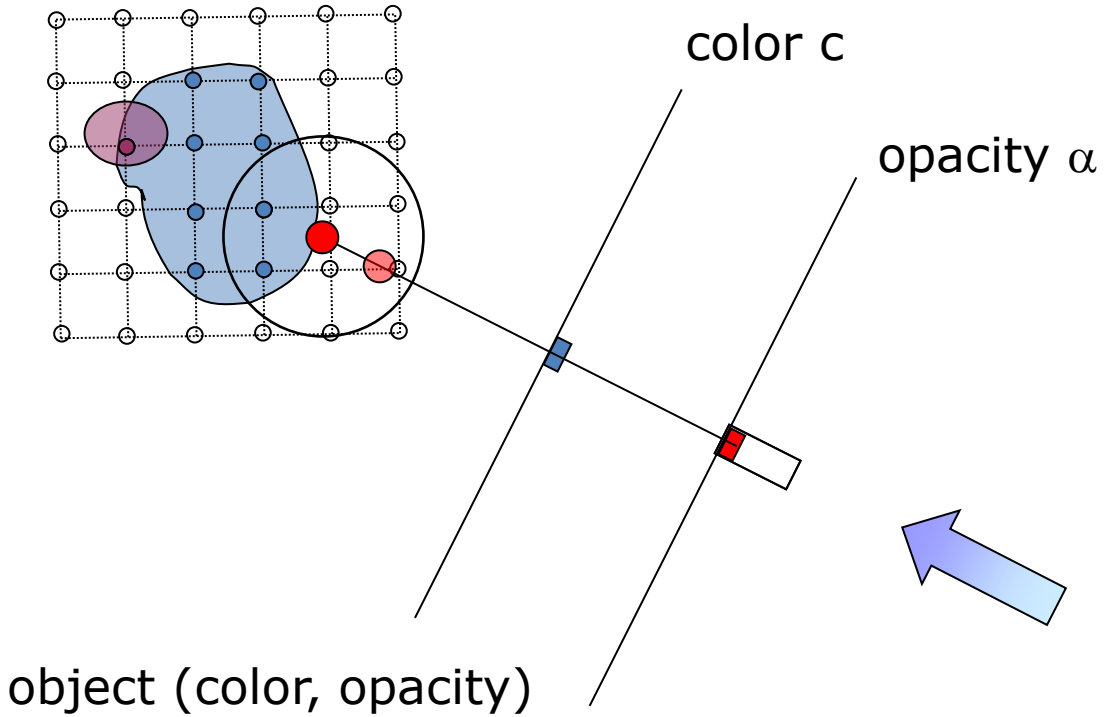
SIEMENS
Ingenuity for life



# Ray casting

SIEMENS
Ingenuity for life

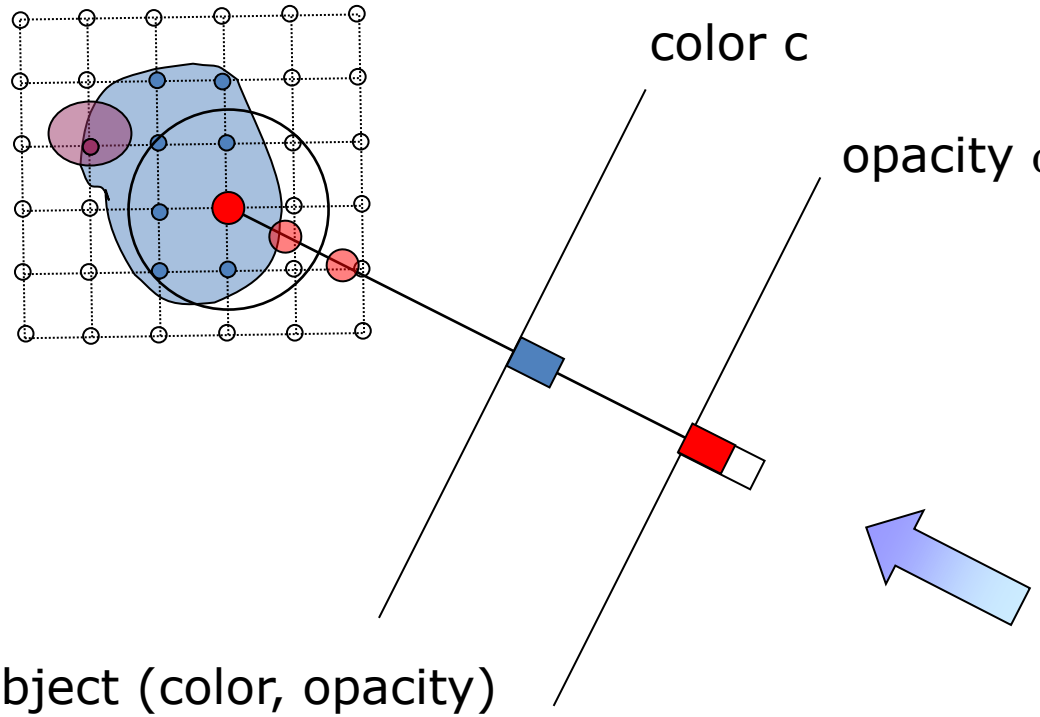
volumetric compositing



# Ray casting

SIEMENS
Ingenuity for life

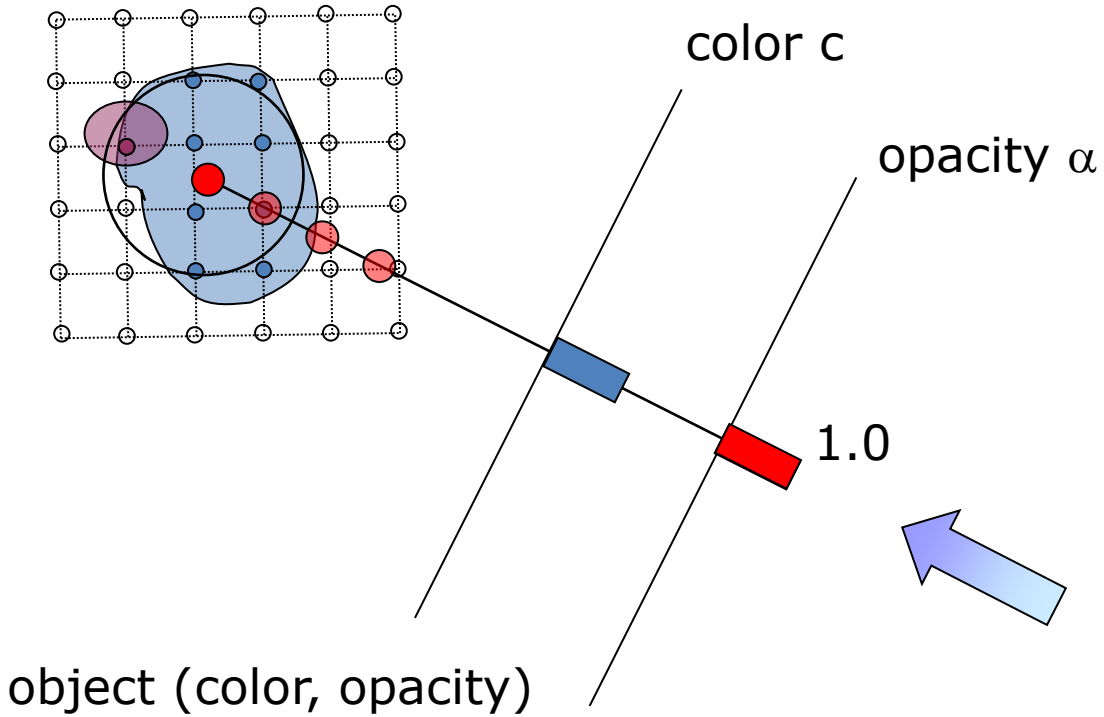
volumetric compositing



# Ray casting

SIEMENS
Ingenuity for life

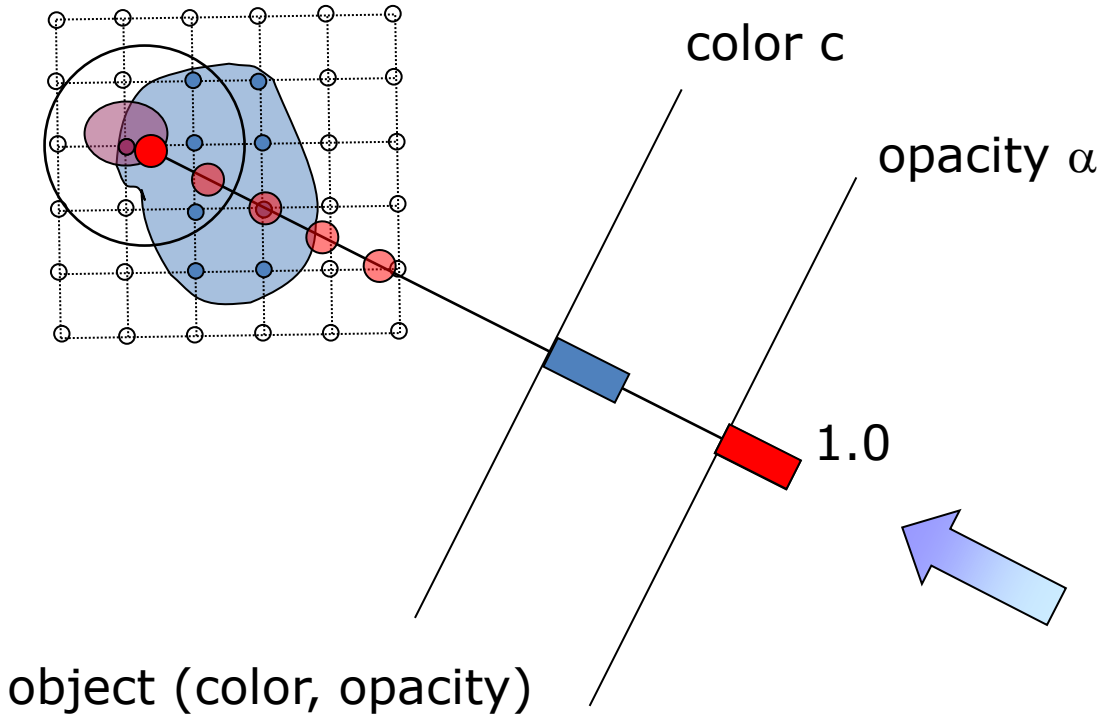
volumetric compositing



# Ray casting

SIEMENS
Ingenuity for life

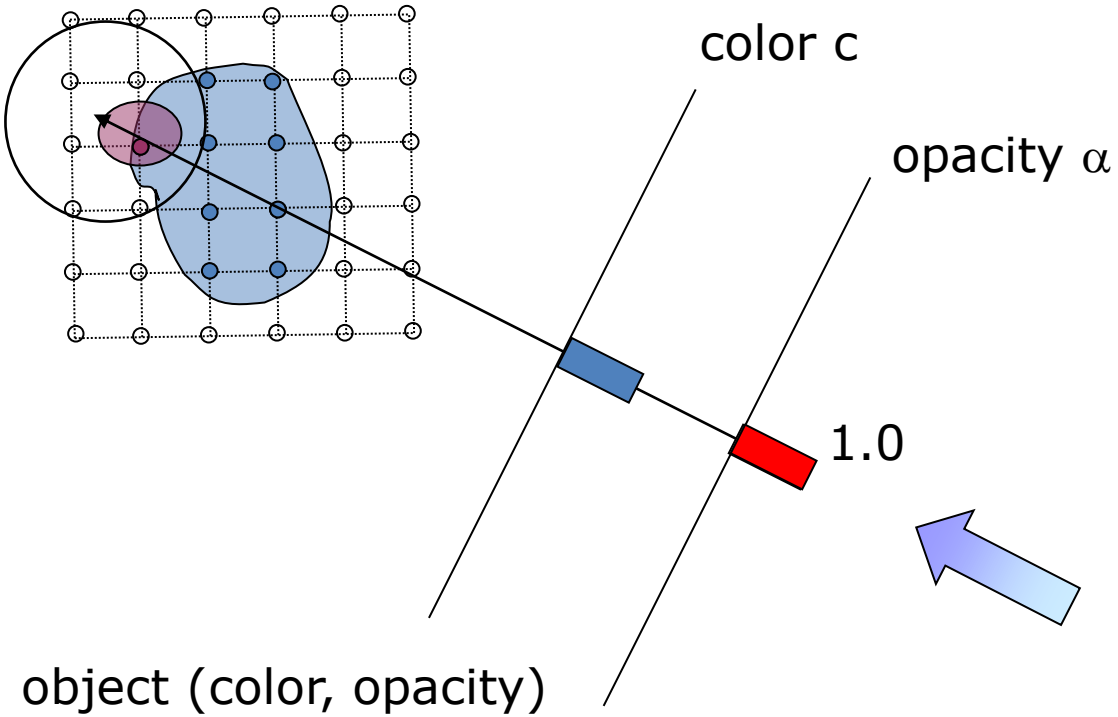
volumetric compositing



# Ray casting

SIEMENS
Ingenuity for life

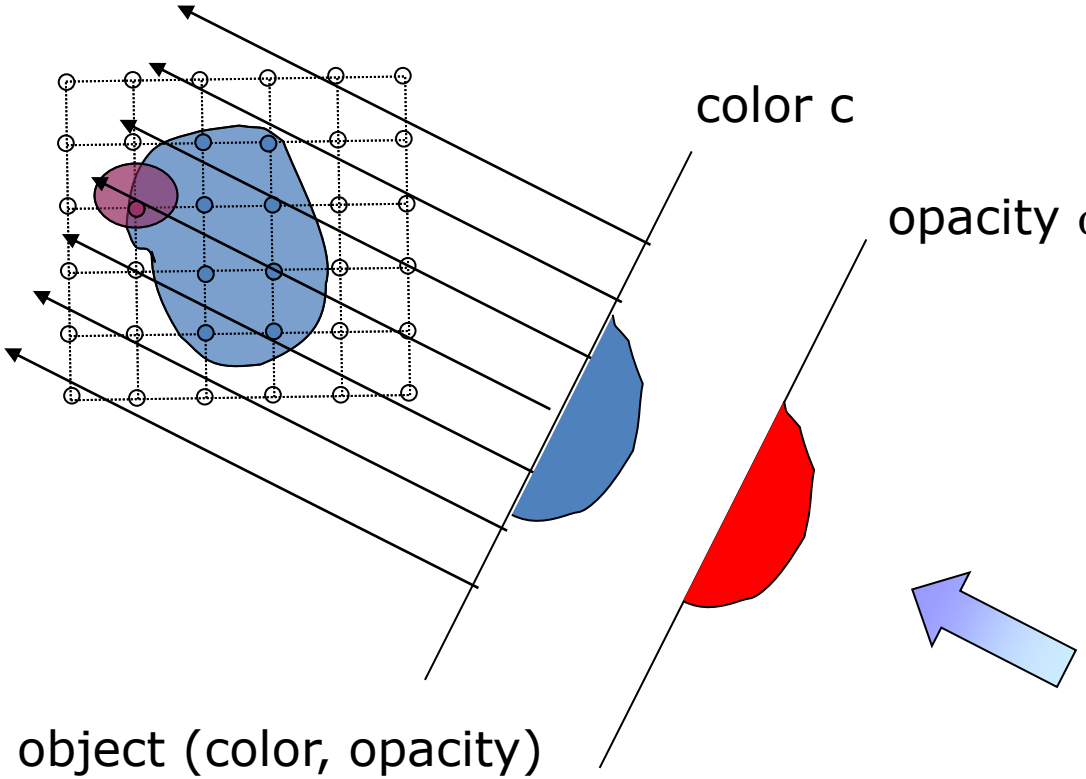
volumetric compositing



# Ray casting

SIEMENS
Ingenuity for life

volumetric compositing



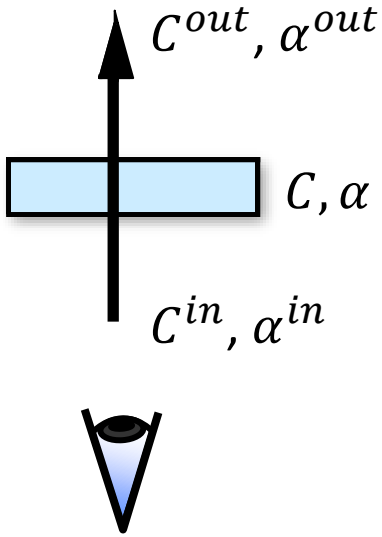
# Ray casting

SIEMENS
Ingenuity for life

- Compositing of semi-transparent voxels
  - Physical model: emissive gas with color $C$ and opacity $\alpha$
  - Front-to-back strategy,
    starts with $C^{in} = (0,0,0)$ and $\alpha^{in} = 0$

$$C^{out} = C^{in} + \left(1 - \alpha^{in}\right) \alpha C$$

$$\alpha^{out} = \alpha^{in} + \left(1 - \alpha^{in}\right) \alpha$$




front-to-back strategy

# Ray casting

- Example

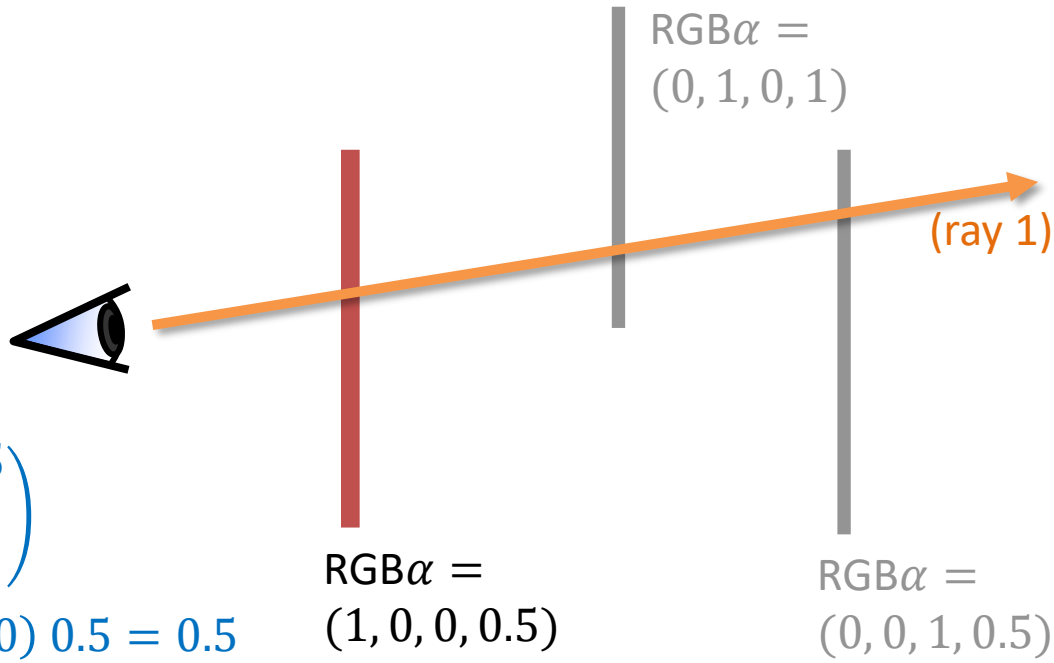
  $\alpha = 1$ → completely opaque
  $\alpha = 0$ → completely transparent

**Ray 1:** $C_{in} = (0,0,0)^{\mathrm{T}} \;\; \alpha_{in} = 0$

$$C_{out} = C_{in} + (1 - \alpha_{in})\,\alpha\, C$$

$$= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + (1-0)\,0.5 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 0 \\ 0 \end{pmatrix}$$

$$\alpha_{out} = \alpha_{in} + (1 - \alpha_{in})\,\alpha = 0 + (1-0)\,0.5 = 0.5$$

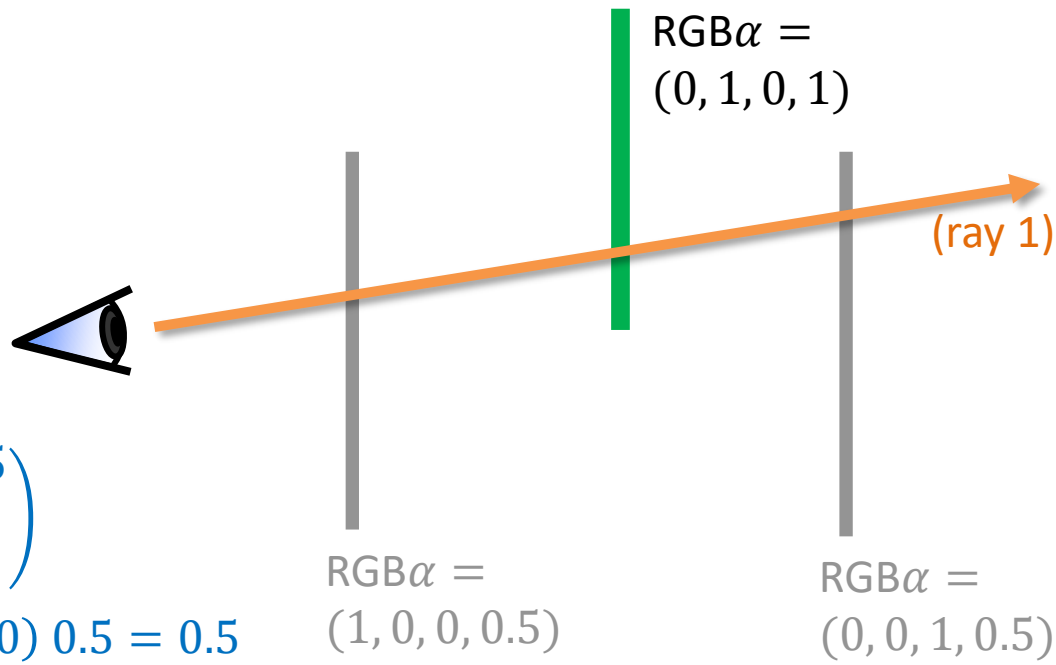


# Ray casting

SIEMENS
Ingenuity for life

- Example

  $\alpha = 1$ → completely opaque
  $\alpha = 0$ → completely transparent




**Ray 1:** $C_{in} = (0,0,0)^{\mathrm{T}} \quad \alpha_{in} = 0$

$$C_{out} = C_{in} + (1 - \alpha_{in})\,\alpha\,C$$

$$= \begin{pmatrix}0\\0\\0\end{pmatrix} + (1-0)\,0.5 \begin{pmatrix}1\\0\\0\end{pmatrix} = \begin{pmatrix}0.5\\0\\0\end{pmatrix}$$

$$\alpha_{out} = \alpha_{in} + (1 - \alpha_{in})\,\alpha = 0 + (1-0)\,0.5 = 0.5$$

$$C_{out} = \begin{pmatrix}0.5\\0\\0\end{pmatrix} + (1-0.5)\,1 \begin{pmatrix}0\\1\\0\end{pmatrix} = \begin{pmatrix}0.5\\0.5\\0\end{pmatrix}$$

$$\alpha_{out} = 0.5 + (1-0.5)\,1 = 1$$

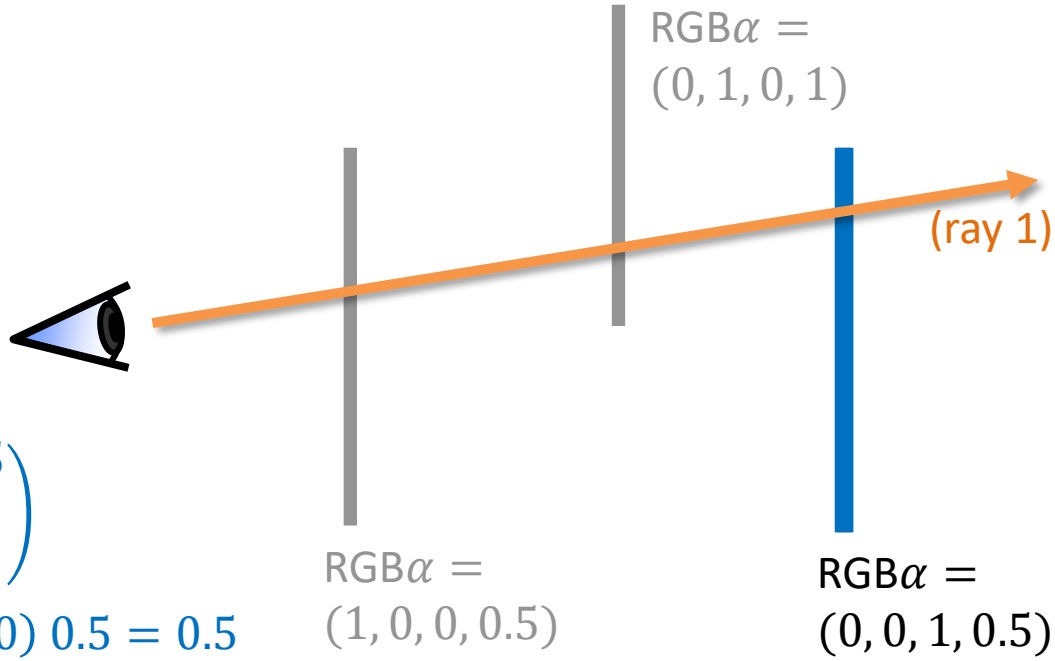
**Early ray termination:**
Stop calculation when $\alpha_{out} \approx 1$

# Ray casting

SIEMENS
Ingenuity for life

- Example

  $\alpha = 1$ → completely opaque
  $\alpha = 0$ → completely transparent




**Ray 1:** $C_{in} = (0,0,0)^T \quad \alpha_{in} = 0$

$$C_{out} = C_{in} + (1 - \alpha_{in})\,\alpha\,C$$

$$= \begin{pmatrix}0\\0\\0\end{pmatrix} + (1-0)\,0.5 \begin{pmatrix}1\\0\\0\end{pmatrix} = \begin{pmatrix}0.5\\0\\0\end{pmatrix}$$

$$\alpha_{out} = \alpha_{in} + (1 - \alpha_{in})\,\alpha = 0 + (1-0)\,0.5 = 0.5$$

$$C_{out} = \begin{pmatrix}0.5\\0\\0\end{pmatrix} + (1-0.5)\,1 \begin{pmatrix}0\\1\\0\end{pmatrix} = \begin{pmatrix}0.5\\0.5\\0\end{pmatrix}$$

$$\alpha_{out} = 0.5 + (1-0.5)\,1 = 1$$

$$C_{out} = \begin{pmatrix}0.5\\0.5\\0\end{pmatrix} + (1-1)\,0.5 \begin{pmatrix}0\\0\\1\end{pmatrix} = \begin{pmatrix}0.5\\0.5\\0\end{pmatrix}$$

$$\alpha_{out} = 1 + (1-1)\,0.5 = 1$$

**Early ray termination:**

Stop calculation when $\alpha_{out} \approx 1$

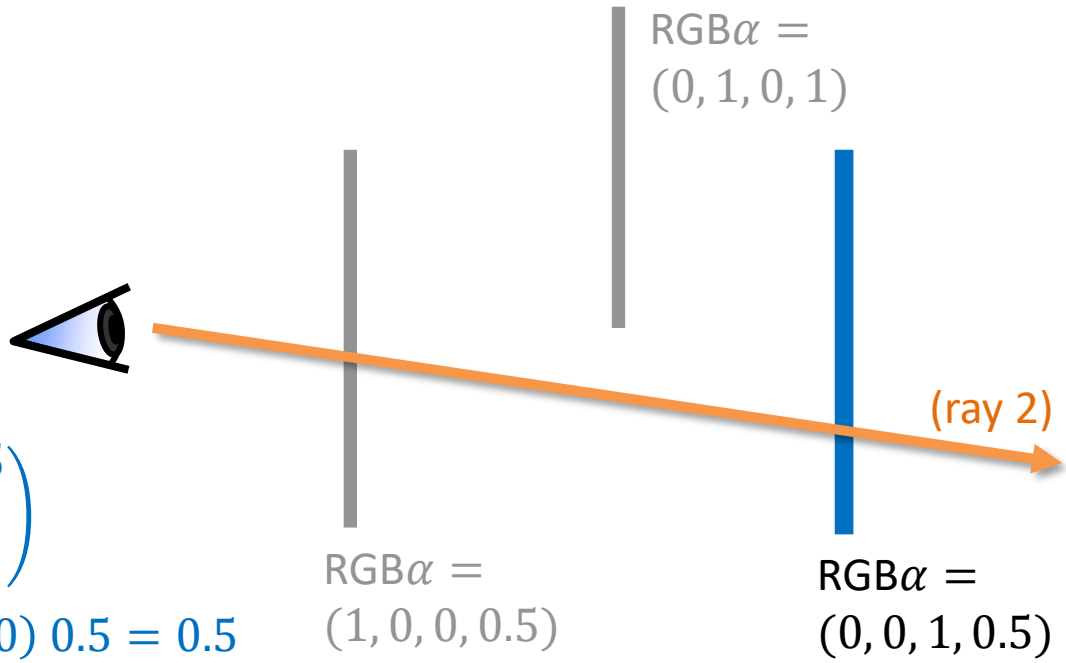
Last object does not affect result since $\alpha_{in} = 1$

# Ray casting

SIEMENS
Ingenuity for life

- Example

  $\alpha = 1$ → completely opaque
  $\alpha = 0$ → completely transparent




**Ray 2:** $C_{in} = (0,0,0)^{\mathrm{T}} \;\; \alpha_{in} = 0$

$$C_{out} = C_{in} + (1 - \alpha_{in})\, \alpha\, C$$

$$= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + (1-0)\, 0.5 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 0 \\ 0 \end{pmatrix}$$

$$\alpha_{out} = \alpha_{in} + (1 - \alpha_{in})\, \alpha = 0 + (1-0)\, 0.5 = 0.5$$

$$C_{out} = \begin{pmatrix} 0.5 \\ 0 \\ 0 \end{pmatrix} + (1-0.5)\, 0.5 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 0 \\ 0.25 \end{pmatrix}$$

$$\alpha_{out} = 0.5 + (1-0.5)\, 0.5 = 0.75$$

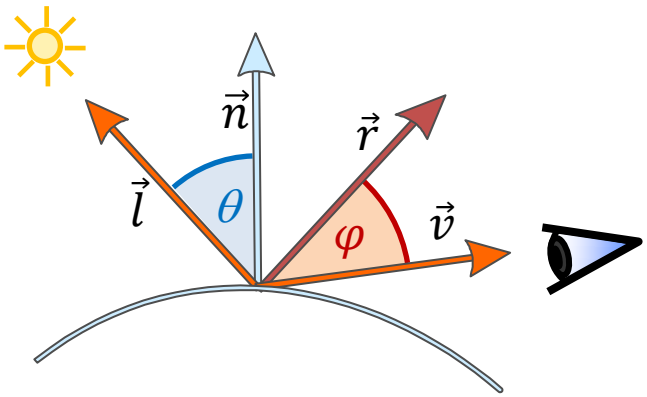
# Lighting

SIEMENS
Ingenuity for life

- Evaluate Phong's illumination model based on…
  - position of current sample and light source
  - samples' color emission assigned by transfer function
  - sample's normal / gradient (e.g., central differences)

DVR without shading

Phong shading



# Ray casting

SIEMENS
Ingenuity for life

- Gradient estimation
  - The gradient vector (normal) is the first-order derivative of the scalar field

$$\nabla f(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial}{\partial x} f(\boldsymbol{x}) \\ \frac{\partial}{\partial y} f(\boldsymbol{x}) \\ \frac{\partial}{\partial z} f(\boldsymbol{x}) \end{pmatrix}$$

partial derivative in x-direction

partial derivative in y-direction

partial derivative in z-direction

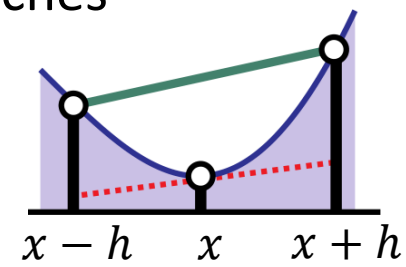
  - We can estimate the gradient using finite differencing schemes (e.g., forward or central differences)

# Ray casting

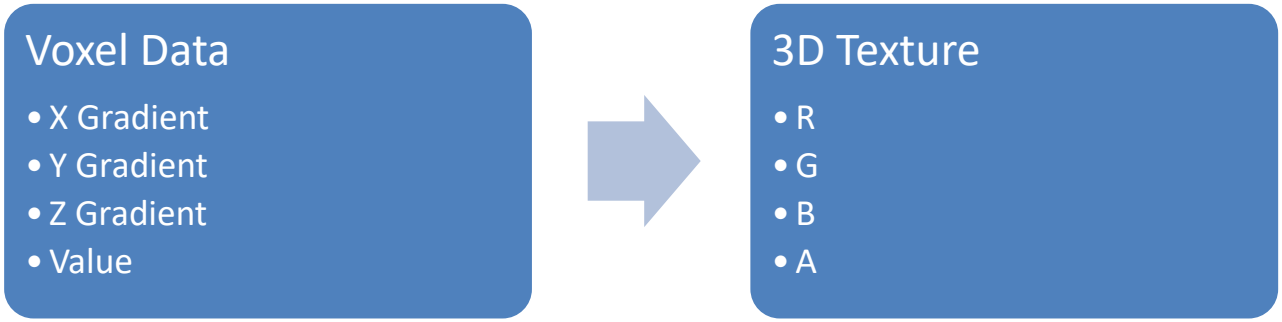
SIEMENS
Ingenuity for life

- Gradient estimation
  - Compute on-the-fly within fragment shader
    - Central differences require 6 additional texture fetches

$$\nabla f(x,y,z) \approx \frac{1}{2h}\begin{pmatrix} f(x+h,y,z) - f(x-h,y,z) \\ f(x,y+h,z) - f(x,y-h,z) \\ f(x,y,z+h) - f(x,y,z-h) \end{pmatrix}$$




  - Pre-compute and store together with voxel data
    - Pack into 3D RGBA texture for GPU-based rendering
    - One lookup to get interpolated gradient + value
    - Requires more memory → not applicable for large volume data



# Ray casting

SIEMENS
Ingenuity for life

- Gradient estimation

Pre-computed,
8 bit per component

on-the-fly
computation

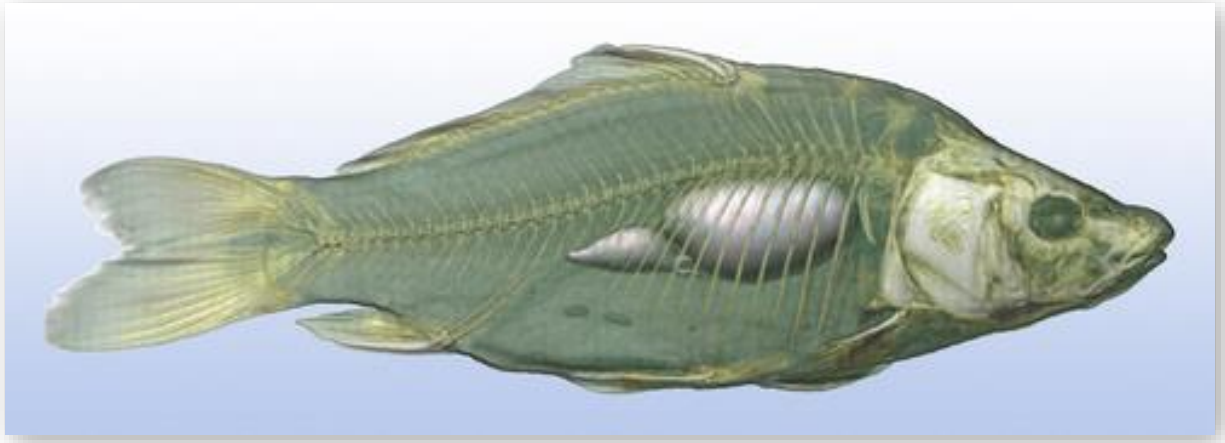
# Ray casting

SIEMENS
Ingenuity for life

- Examples







# Compositing Schemes

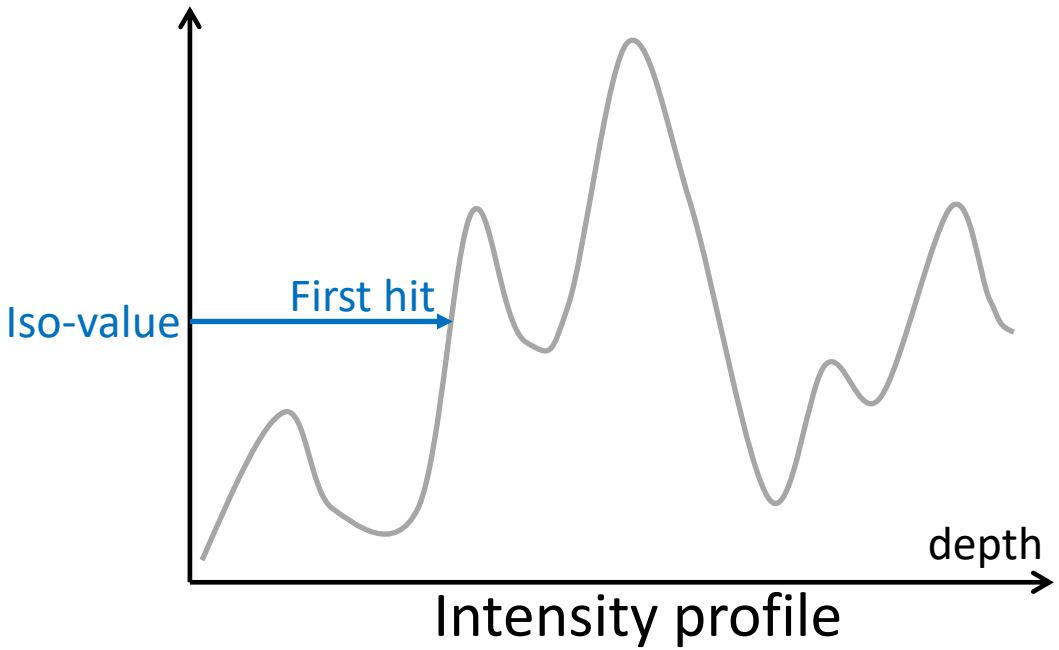
SIEMENS
Ingenuity for life

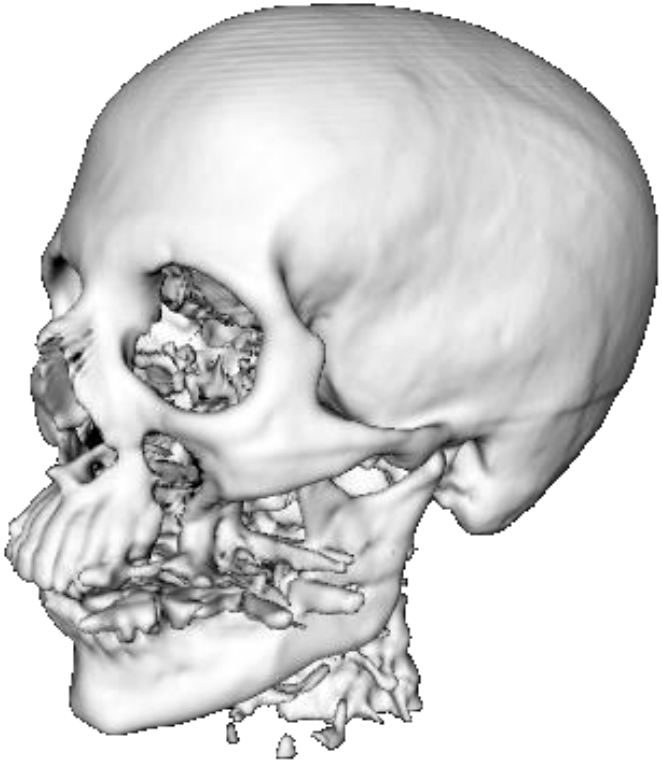
- Variations of compositing schemes
  - Surface rendering / first hit: stop ray traversal if an iso-surface is hit, and shade the surface points
  - Average: simply accumulate colors but do not account for opacity
  - Maximum: only takes the maximum color along the ray and displays it

# Compositing Schemes

SIEMENS
Ingenuity for life

- Surface rendering / first hit
  - Stop ray traversal if an iso-surface is hit, and shade the surface points
  - Produces same result as marching cubes, but with higher accuracy

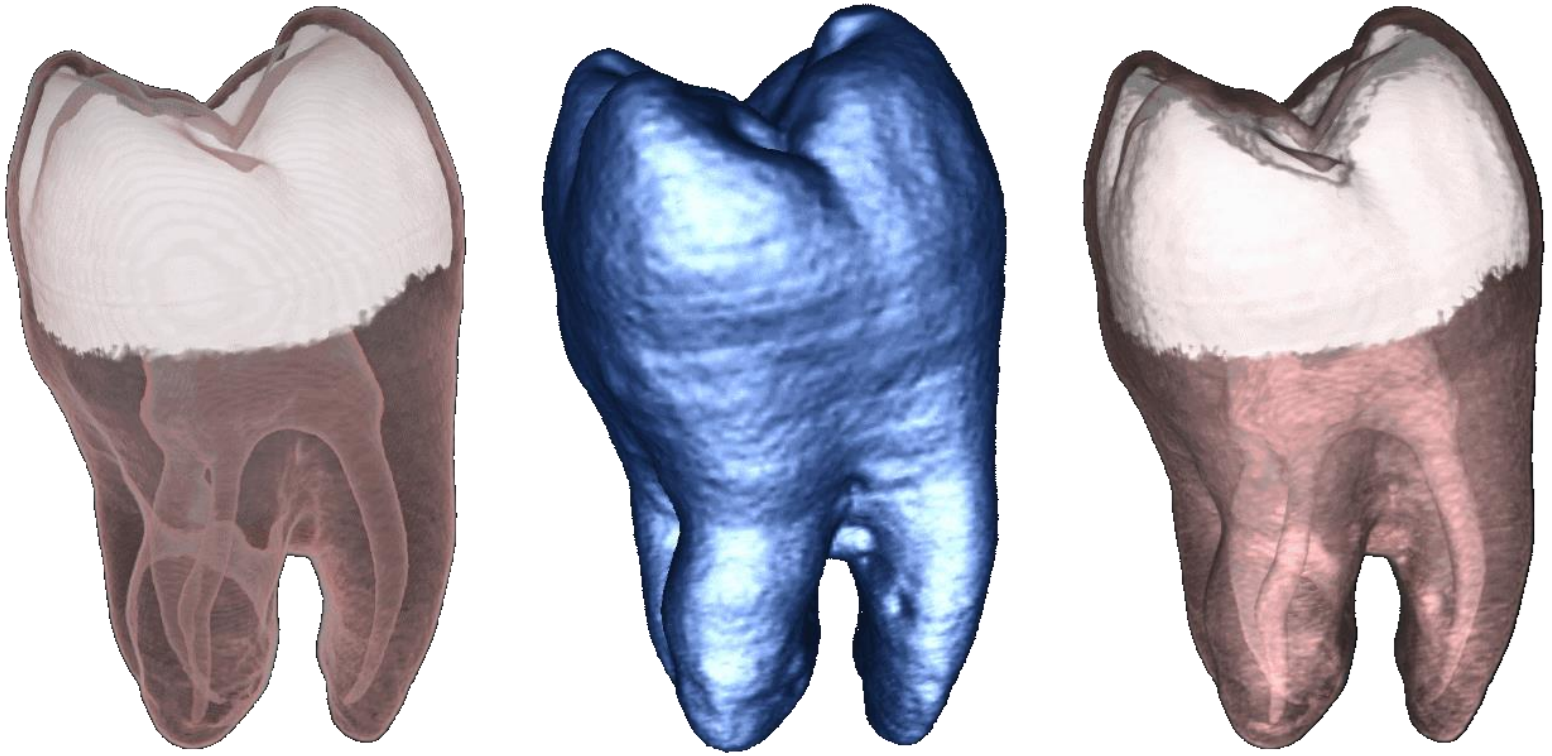


# Compositing Schemes

SIEMENS
Ingenuity for life

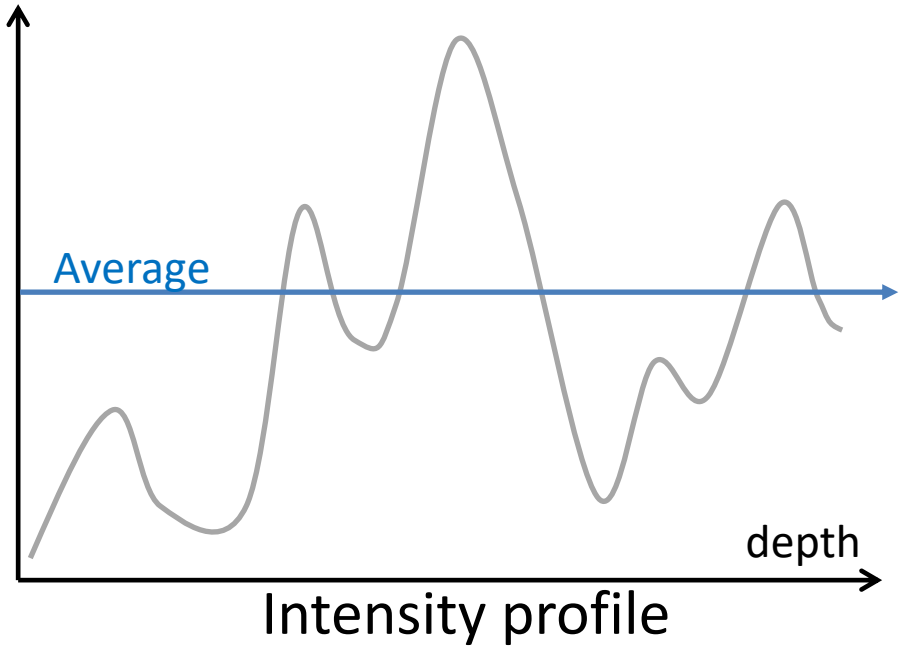
- Compositing: First hit (middle image)



# Compositing Schemes

SIEMENS
Ingenuity for life

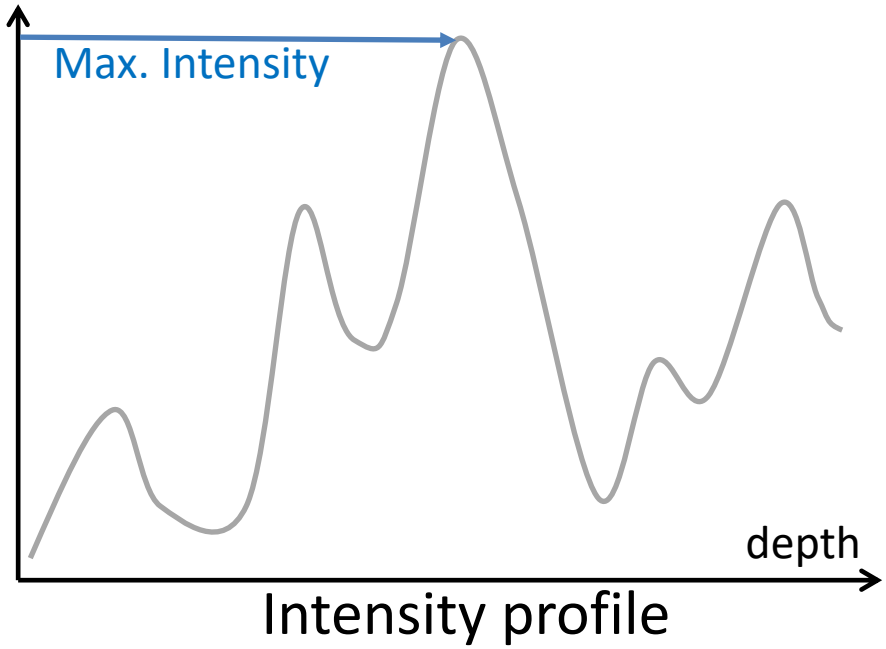
- Compositing: Average
  - Accumulate colors but do not account for opacity
  - Produces an X-ray image




# Compositing Schemes

SIEMENS
Ingenuity for life

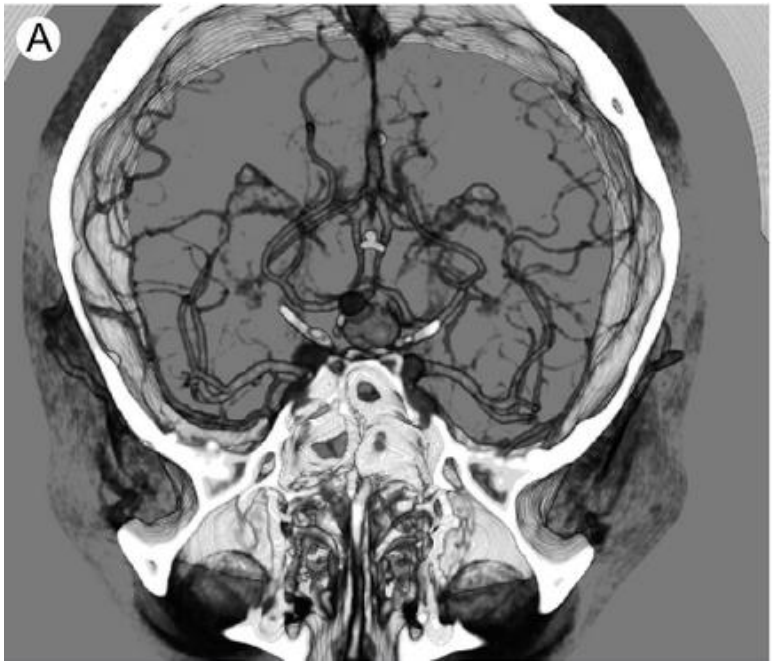
- Maximum Intensity Projection (MIP)
  - Only takes the maximum color along the ray and displays it



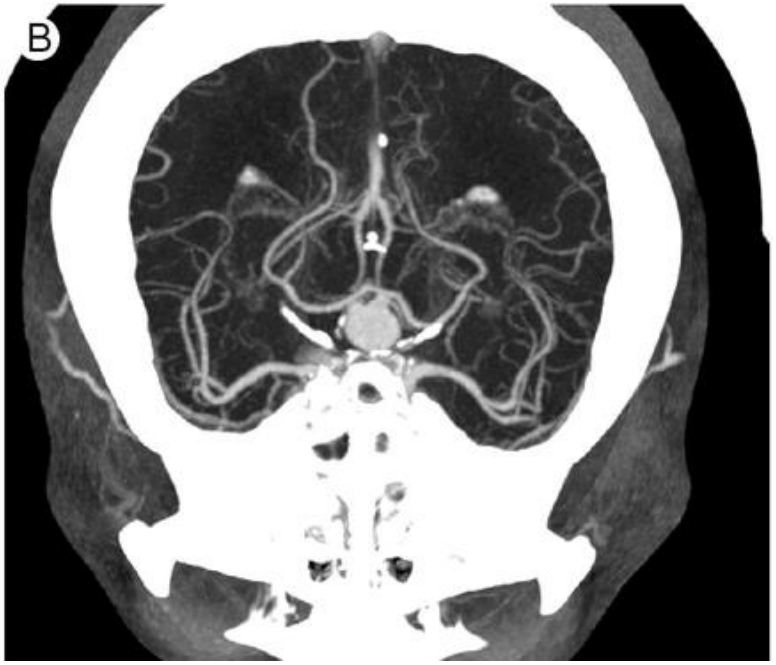

# Compositing Schemes

SIEMENS
Ingenuity for life

- Maximum Intensity Projection (MIP)
  - Often used for magnetic resonance angiograms
  - Good to extract vessel structures




Direct volume rendering

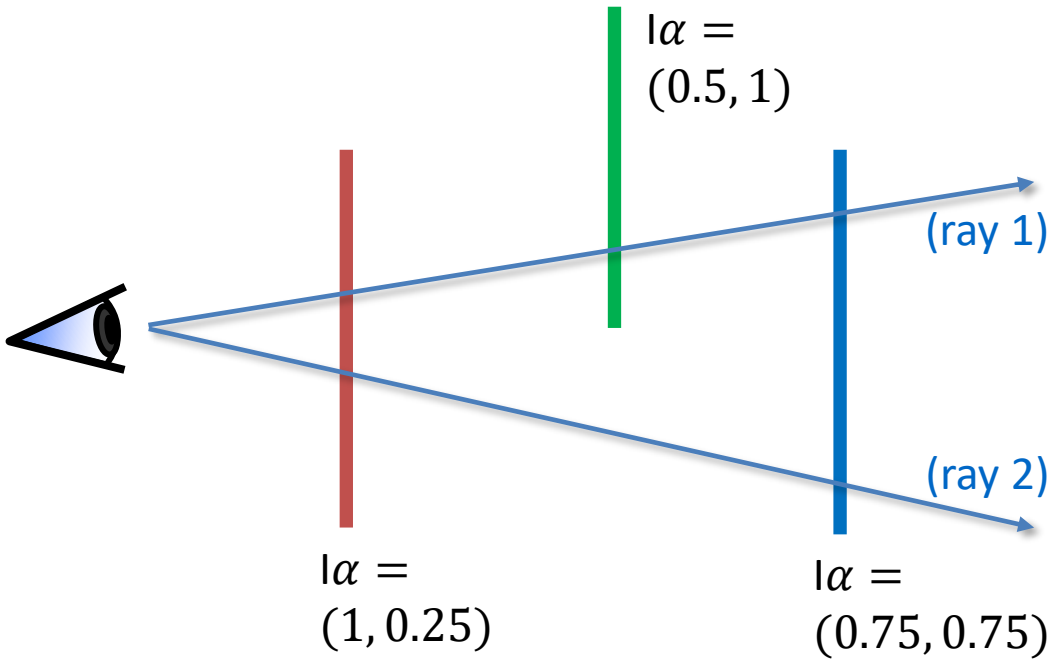



Maximum intensity projection

# Ray casting

- Example

$I$… intensity
$\alpha = 1$ → completely opaque
$\alpha = 0$ → completely transparent




Ray 1 with average compositing:

$$I_{out} = \frac{1}{3}(1 + 0.5 + 0.75) = 0.75$$

Ray 2 with maximum intensity projection:

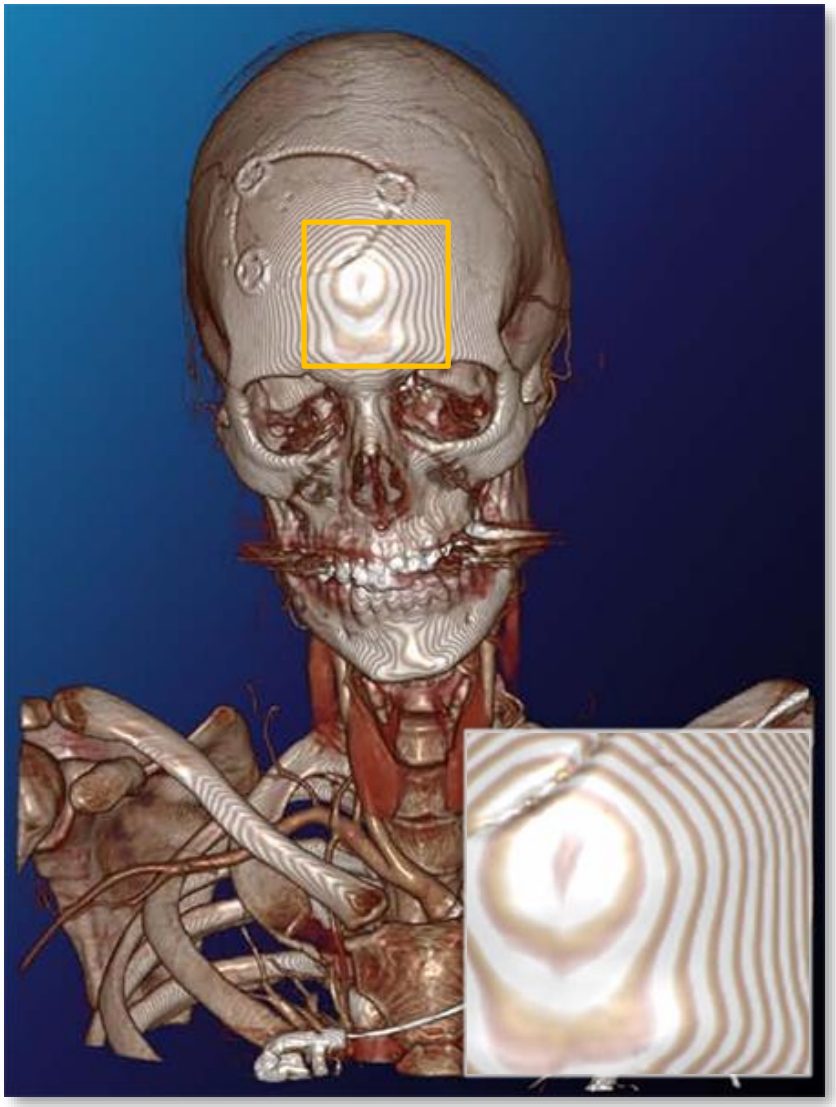
$$I_{out} = \max(1, 0.75) = 1$$

Both average compositing and MIP do not account for opacity

# Ray casting

SIEMENS
Ingenuity for life

- Sampling artifacts
  - To few samples along the ray
  - Solution: Increase sampling rate to Nyquist frequency
    → at least 2 samples per voxel

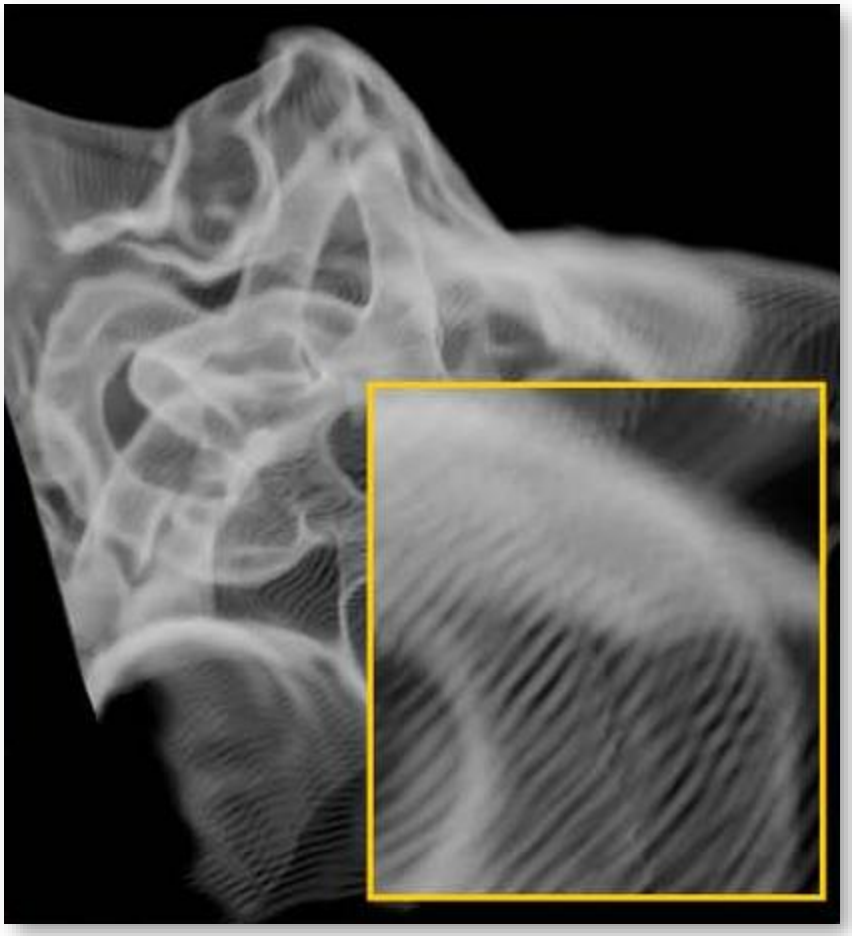


[Engel et al. 06]

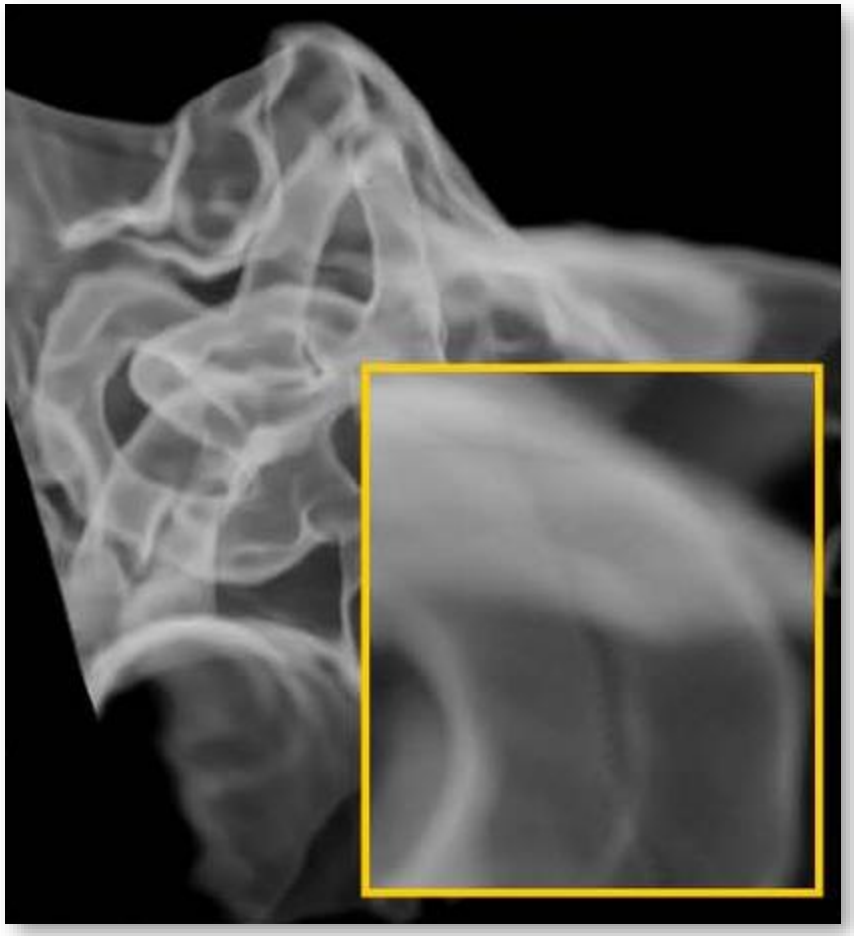
# Ray casting

SIEMENS
Ingenuity for life

- Sampling artifacts



128 samples



284 samples

[Engel et al. 06]

# Ray casting

SIEMENS
Ingenuity for life

- Remove artifacts by stochastic jittering of ray-start position

[Engel et al. 06]

# Summary

SIEMENS
Ingenuity for life

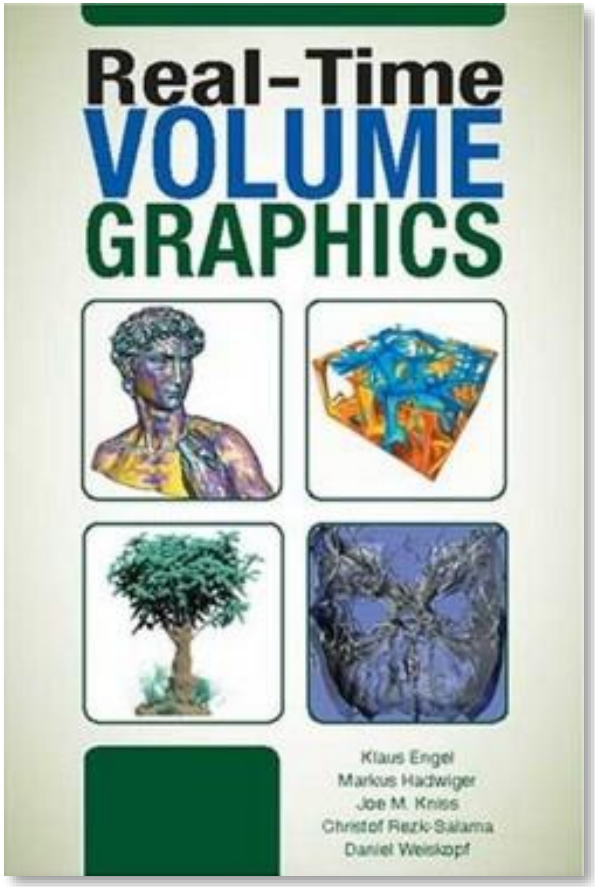
- Surface rendering
  - Indirect representation
  - Conveys surface impression
  - Hardware supported rendering (fast?!)
  - Iso-value-definition
- Direct volume rendering
  - Direct representation
  - Conveys volume impression
  - Often realized in software (slow?!)
  - Transfer function specification

# Further Resources

SIEMENS
Ingenuity for life

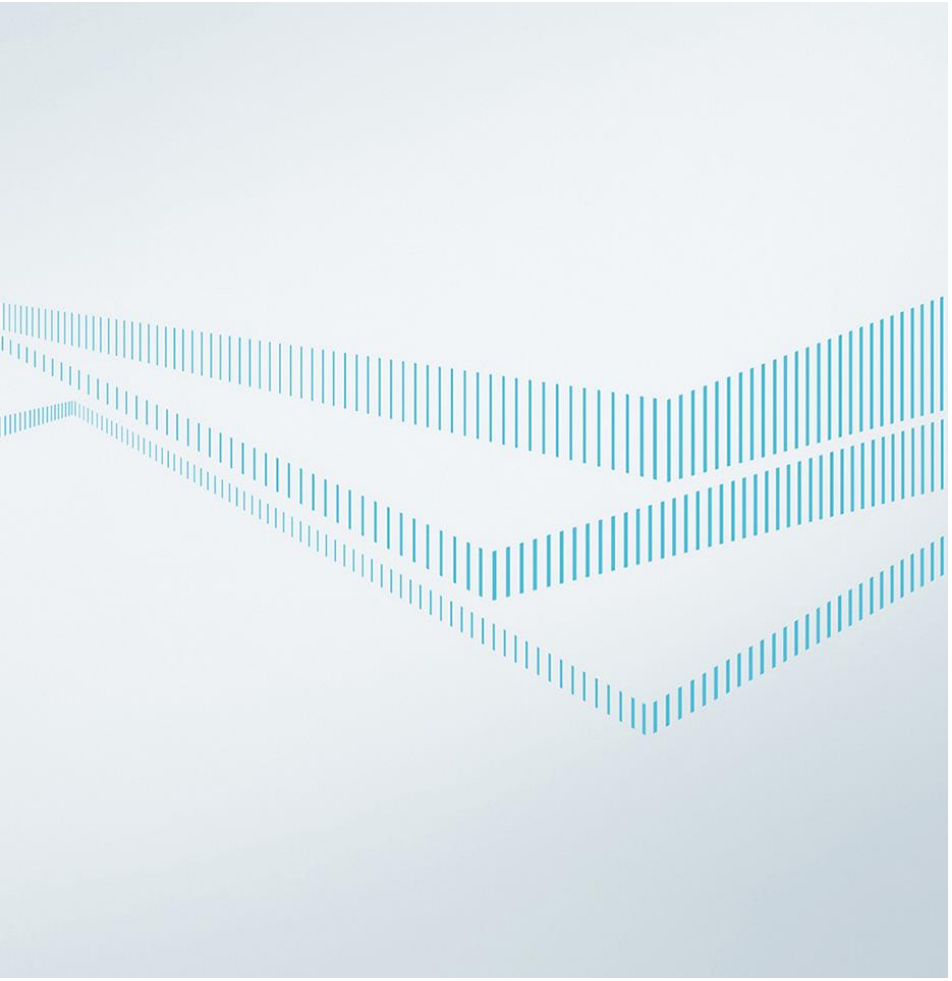
- Engel et al. 06:
  Real-time volume graphics

- Tutorial slides:
  www.real-time-volume-graphics.org



# Contact information

SIEMENS
Ingenuity for life



**Dr. Johannes Kehrer**

Siemens Technology
T RDA BAM IBI-DE
Otto-Hahn-Ring 6
81739 München, Deutschland

E-mail:
kehrer.johannes@siemens.com

Internet
siemens.com/innovation