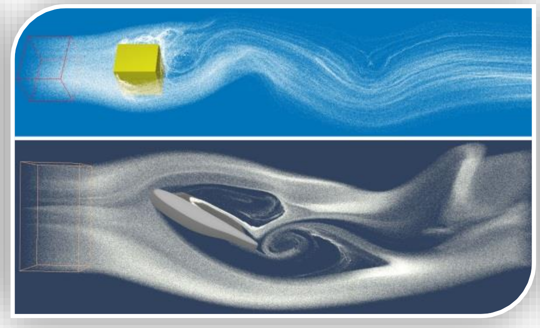
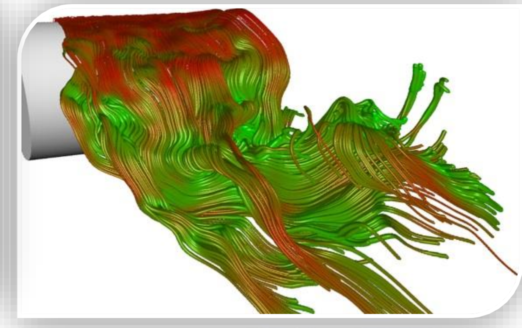
## Master Practical Course

# Interactive Visual Data Analysis





computer graphics & visualization

# About last week

- Tip: Fix Shader-Compiler warnings!

```
struct PSOut {
    float4 Color : SV_Target;
    float Depth : SV_Depth;
};

PSOut psRayCasting(PSIn inFragment) : SV_Target { ...
```

```
warning X3576: semantics in type overridden by variable/function or enclosing type
```

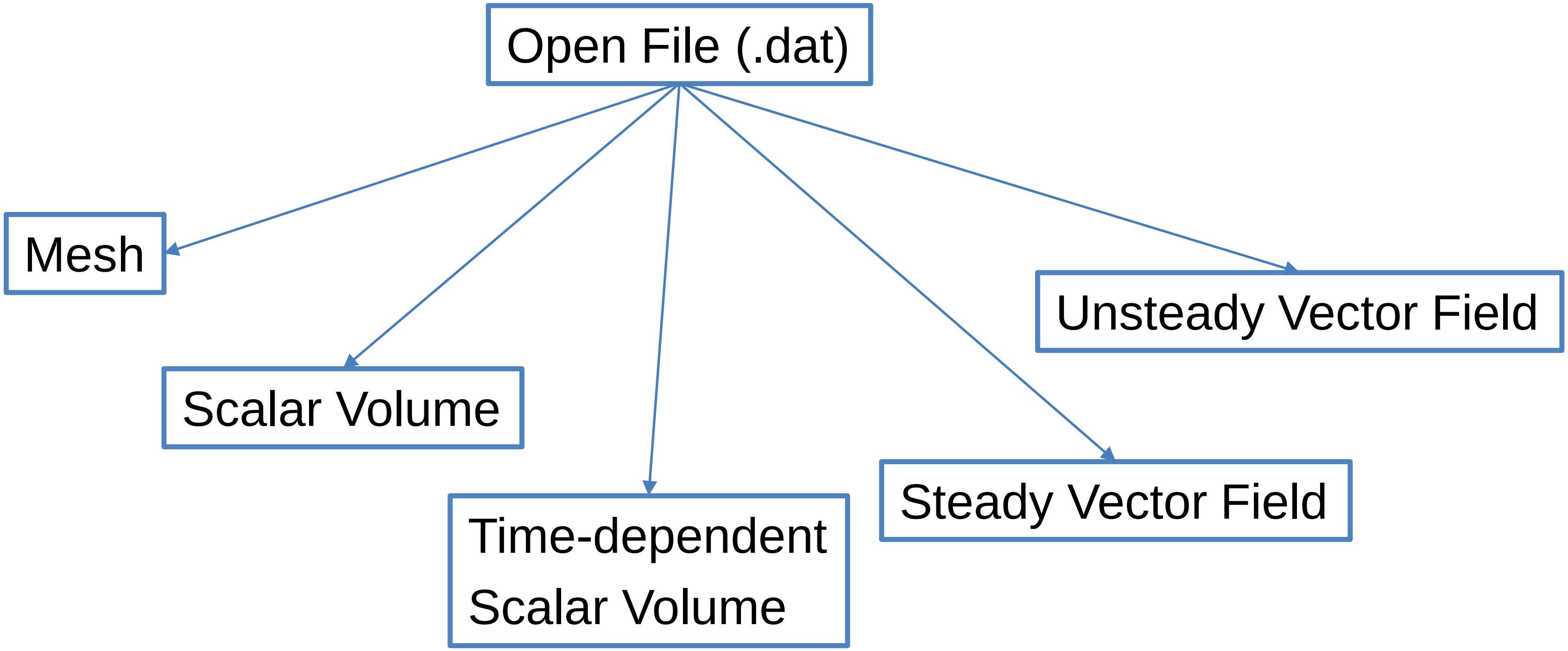
```
float4 v = (1,2,3,4);
```

```
warning X3081: comma expression used where a vector constructor may have been intended
```

```
float4 v = float4(1,2,3,4);
```

# Long-Term Goal

- Visualization tool for different kinds of data
  - Build your application incrementally
  - Keep solutions from past assignments working

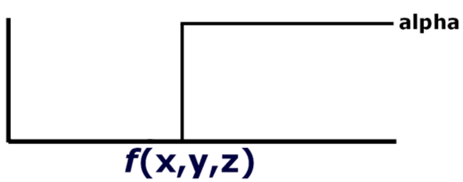


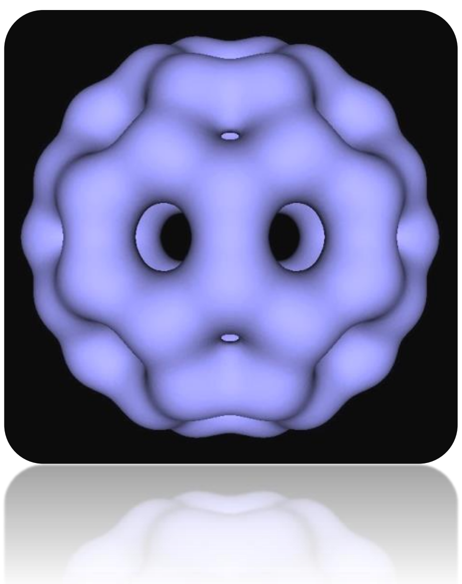
# Today



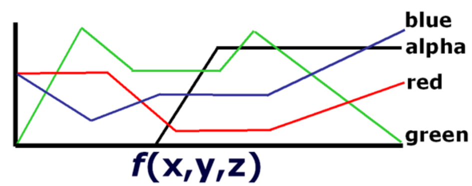
- Assignment 4: Direct Volume Rendering (DVR)

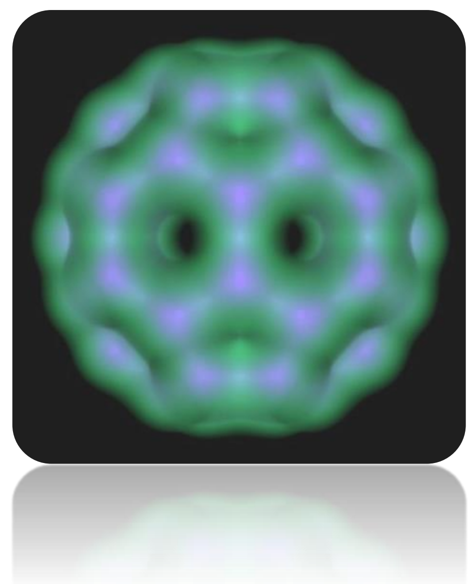
Iso Surface



DVR

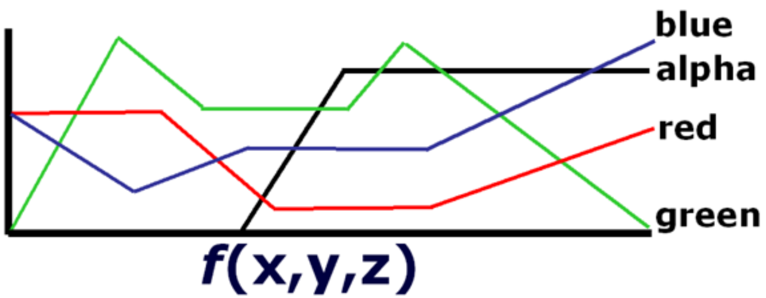


# Let there be color!

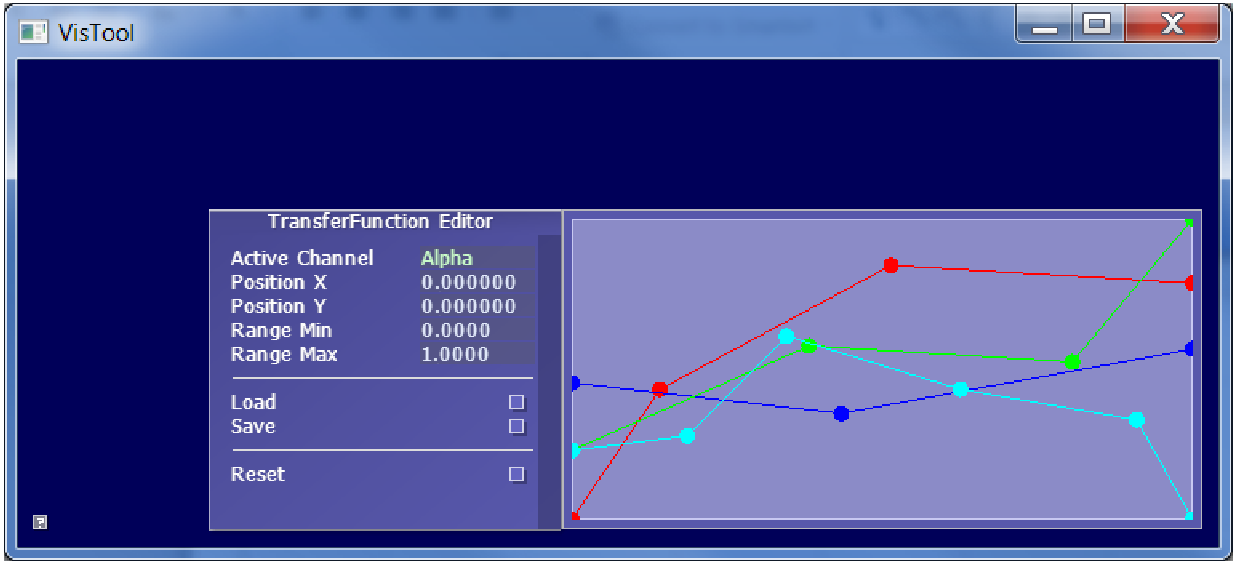


- Transfer functions:
  - Map scalar values f(x,y,z) to rgb color/alpha values



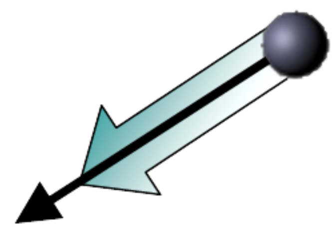

# Transfer Function Editor

- We‘ve got a ready-to-use transfer function editor for you!
- Function editable via AntTweakBar based GUI
- Provides function as 1D RBGA texture

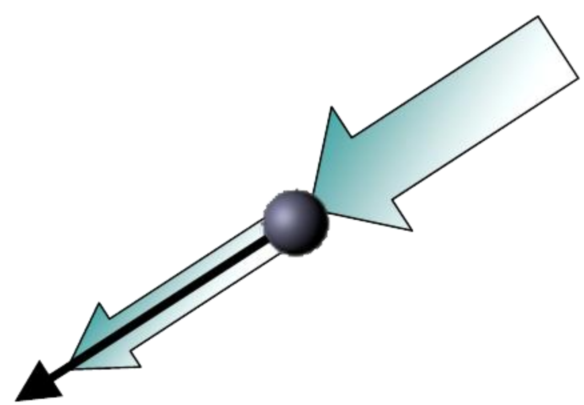


# DVR: Optical Model (1)

- Light (Particle) interaction with density volume
- Emission

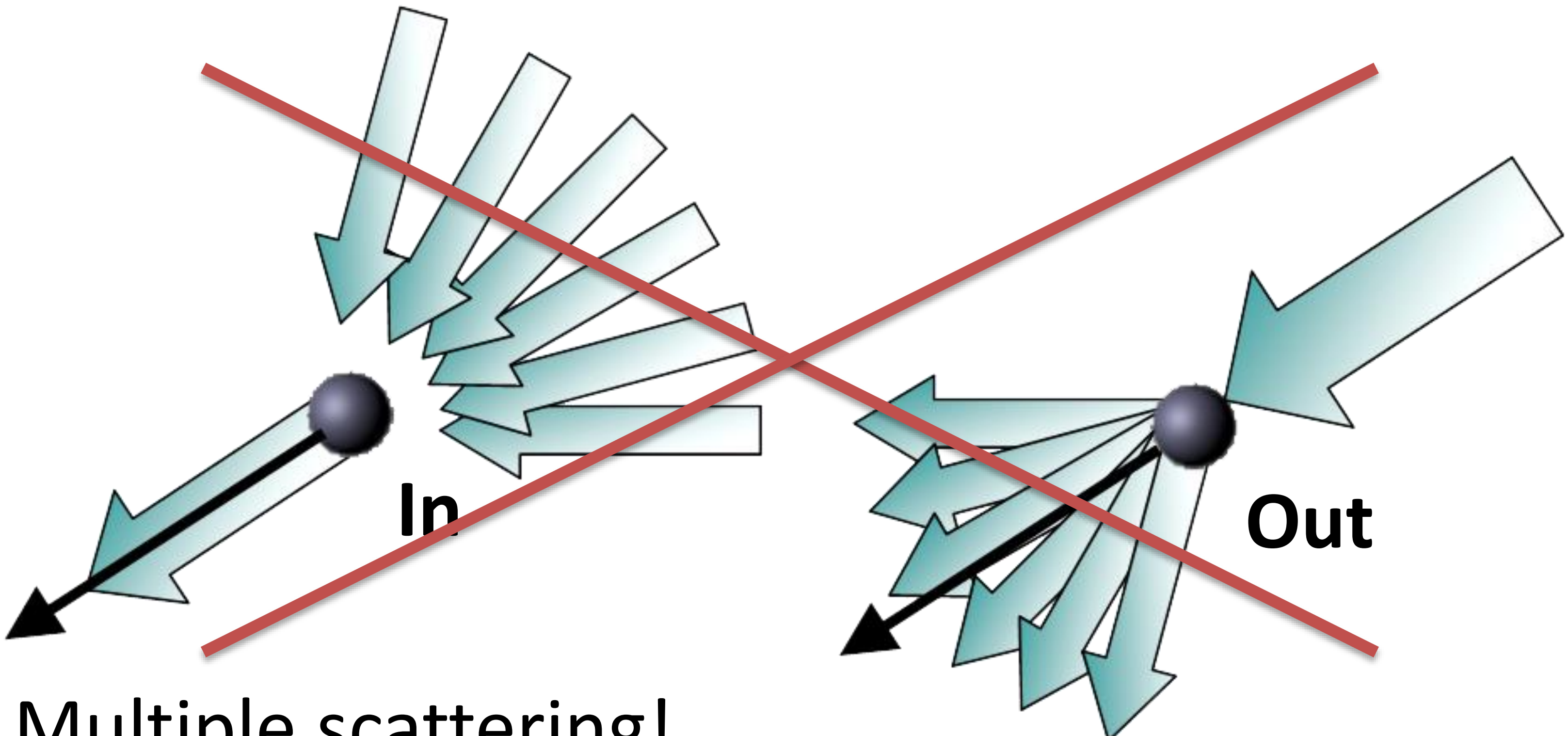


- Absorption



# DVR: Optical Model (2)

- Scattering

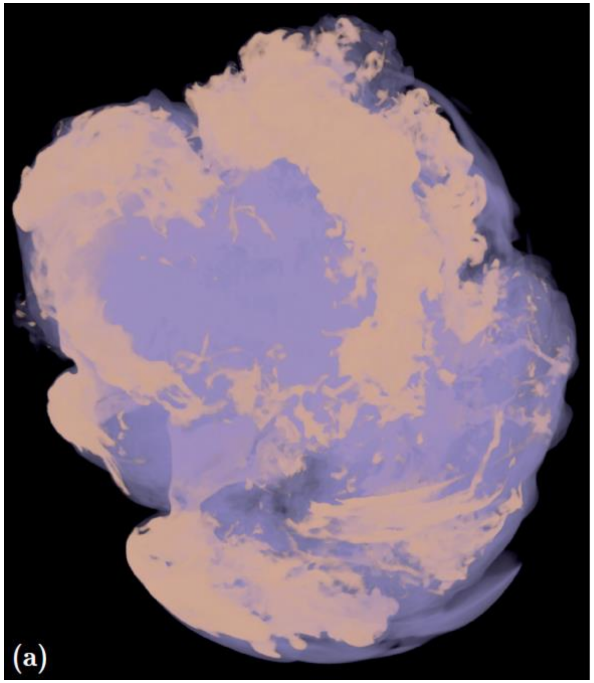



- Multiple scattering!
- Not (immediately) possible in real time

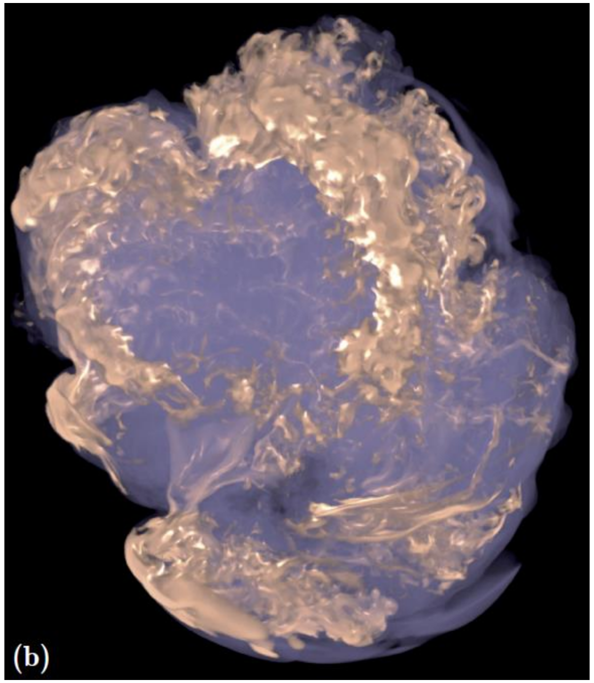
# Example of Recent Work



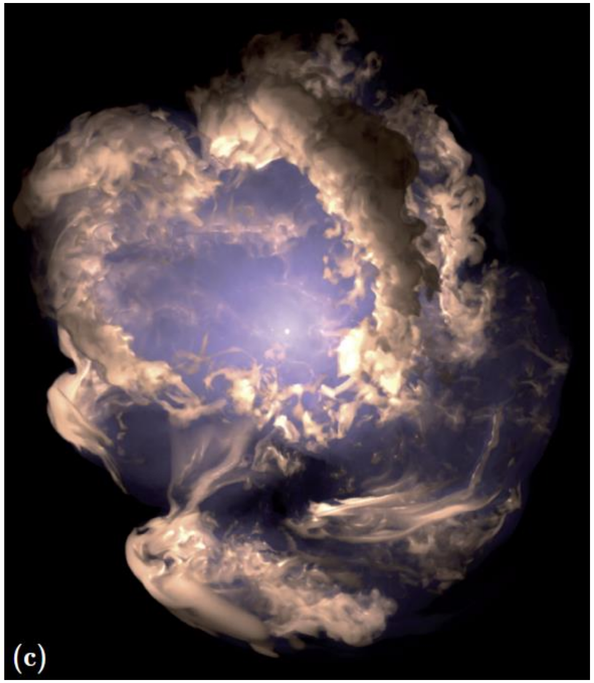
- Realtime approximation of multiple scattering:
  **Ambient Volume Scattering, M. Ament, F. Sadlo, D. Weiskopf**, *IEEE Transactions on Visualization and Computer Graphics*, 19(12): 2936-2945, 2013.




Emission/Absorption



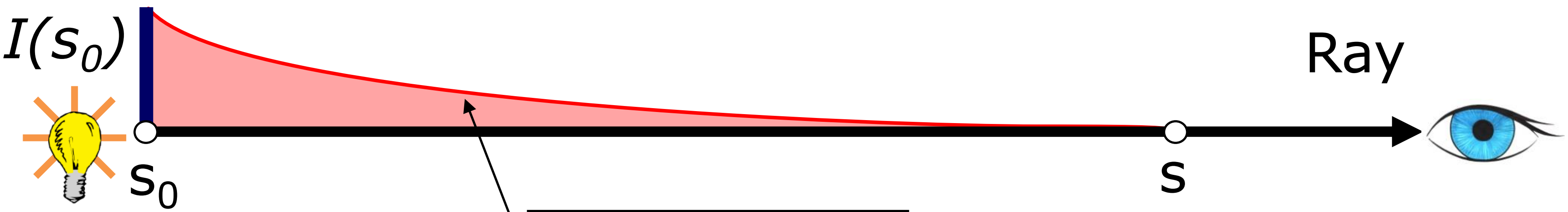

Ambient Occlusion




Ambient Scattering

# DVR: Ray Integration






Initial intensity (Emission) at $s_0$

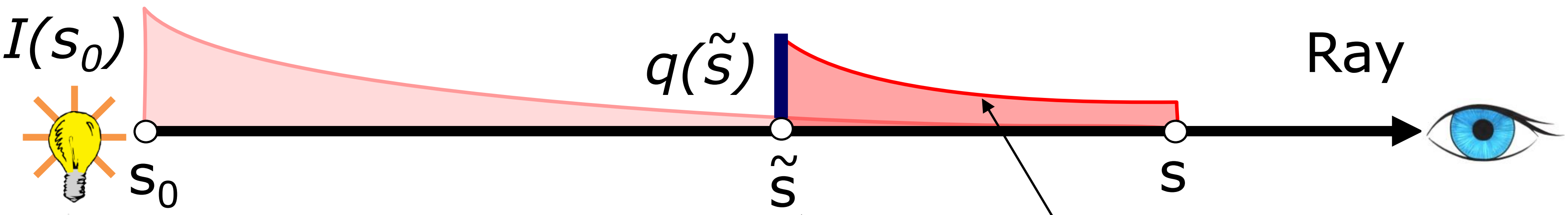
Attenuation along the distance $s_0$ - s

$$I(s) = \boxed{I(s_0)}\boxed{\cdot\, e^{-\tau(s_0,s)}}$$

Optical Depth $\boldsymbol{\tau}$
Absorption $\boldsymbol{\kappa}$

$$\tau(s_1, s_2) = \int_{s_1}^{s_2} \kappa(s)\, ds$$

# DVR: Ray Integration

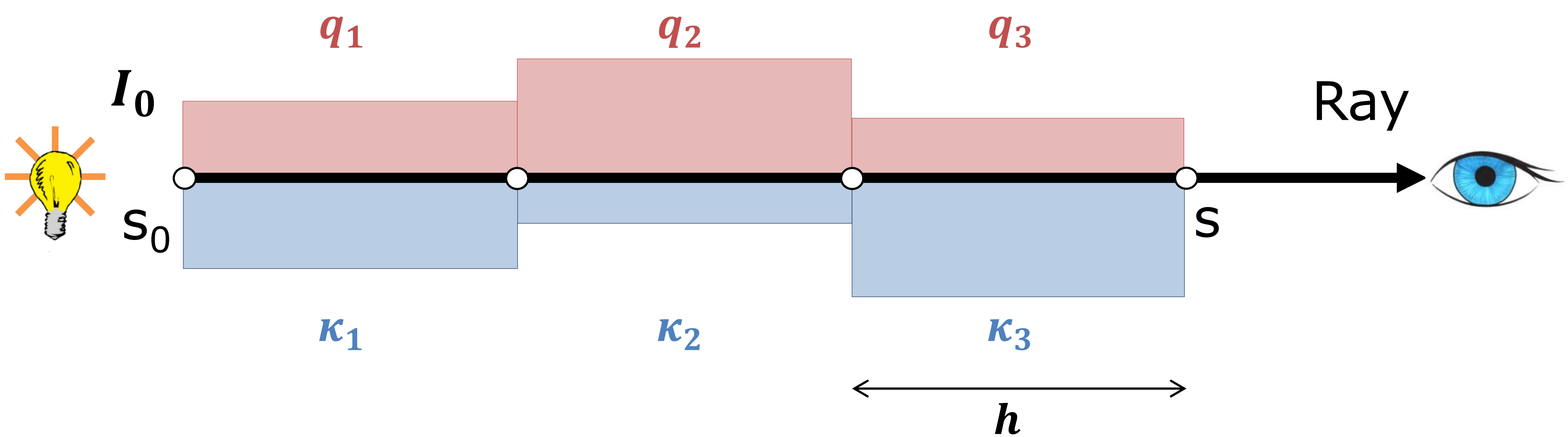



Active Emission at $\tilde{s}$

Attenuation along the distance $\tilde{s}$ - s

$$I(s) = I(s_0) \cdot e^{-\tau(s_0,s)} \quad + \int_{s_0}^{s} \boxed{q(\tilde{s})} \boxed{\cdot\, e^{-\tau(\tilde{s},s)}} \, d\tilde{s}$$

# DVR: Discretization




- Riemann sum
  - Sample at equidistant points
  - Assume constant properties in each segment
  - Some more approximations…
- With $\alpha_i := \kappa_i h$:

$$I(s) \approx q_3\alpha_3 + q_2\alpha_2(1-\alpha_3) + q_1\alpha_1(1-\alpha_2)(1-\alpha_3) + \mathrm{I}_0(1-\alpha_1)(1-\alpha_2)(1-\alpha_3)$$

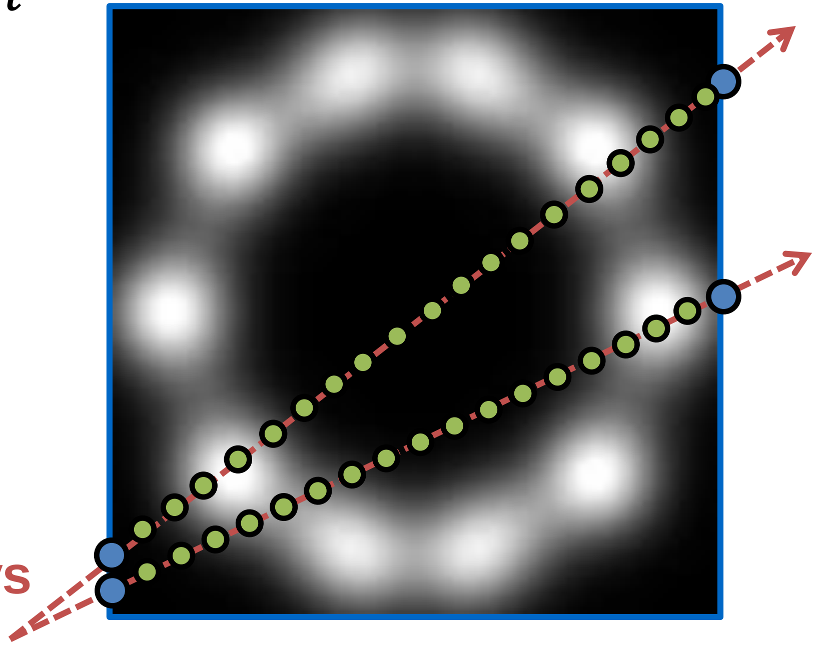
- This is alpha blending! (colors and alphas)

# Implementation (1)

- Step through the volume along view rays
- At each sample point
  - Sample scalar volume
  - Apply transfer function → RGBA value: $(C, \alpha')$
    - $\alpha'$ corresponds to $\kappa$
  - Adjust $\alpha'$ according to step size $h$

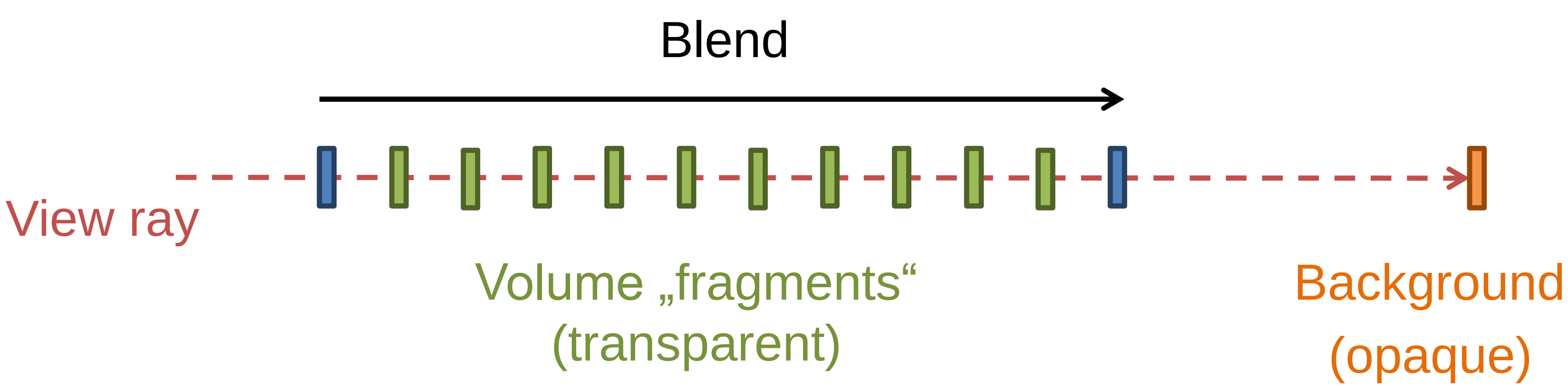
$$\alpha := \alpha' \cdot h$$

  - Result: „Fragment“ $(C, \alpha)$



# Implementation (2)

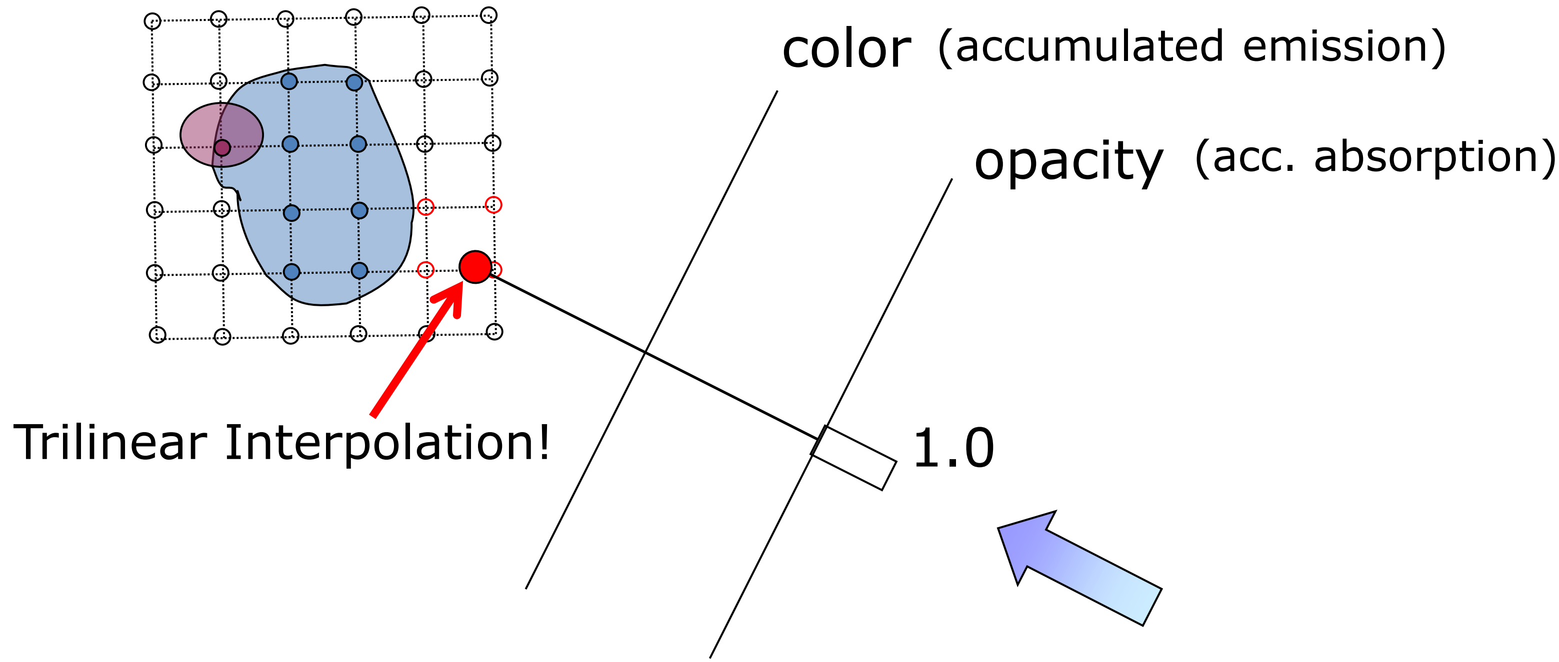
- Accumulate volume „fragments“ using front-to-back alpha blending („blend under“)
  - $\alpha_0 := 0, \quad C_0 := 0$
  $$C_{new}\alpha_{new} = C_F\alpha_F + (1-\alpha_F)C_B\alpha_B$$
  $$\alpha_{new} = \alpha_F + (1-\alpha_F)\alpha_B$$
  - Pre-multiplied alpha: store and update $(C_x\alpha_x, \alpha_x)$ instead of $(C_x, \alpha_x)$
- Terminate if exit point reached **or $\alpha$ close to 1**



# Implementation (3)

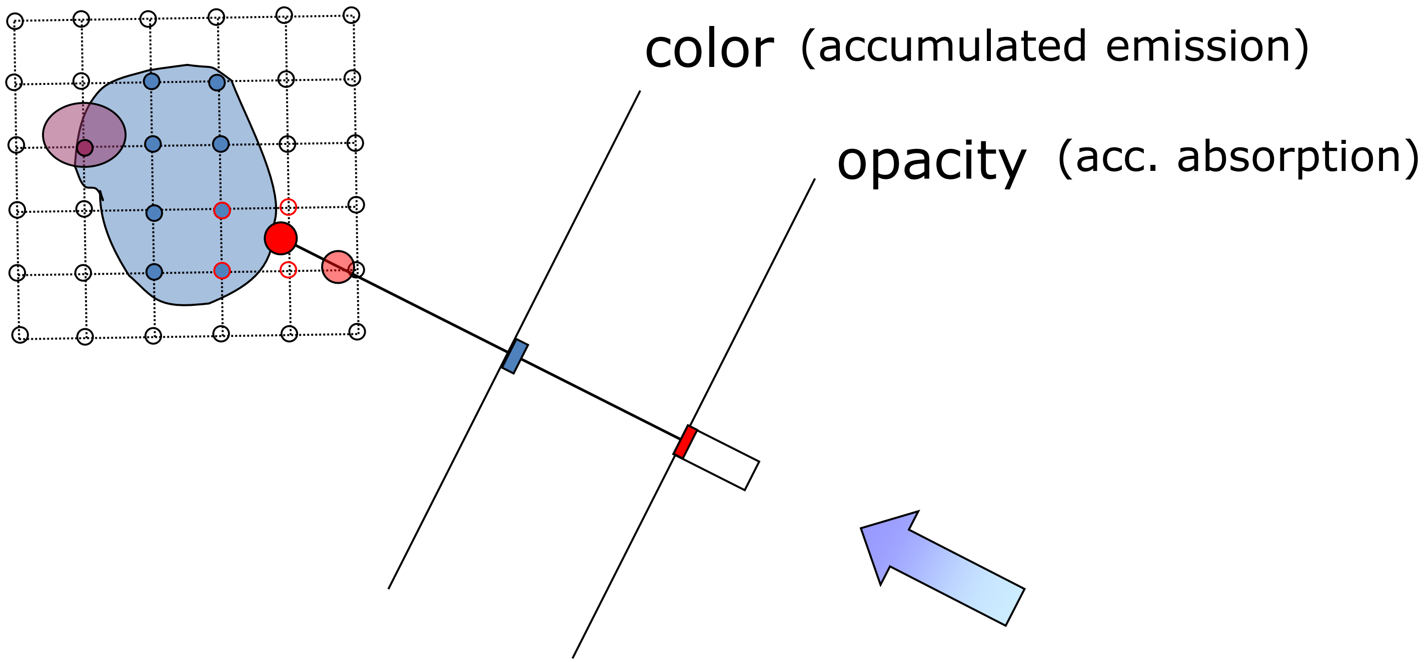
- Result is a single „accumulated“ fragment
- „Blend over“ background (back-to-front alpha-blending)
  - $C_{new} = C_F\alpha_F + (1-\alpha_F)C_B$
    $$\alpha_{new} = \alpha_F + (1-\alpha_F)\alpha_B$$
  - Configure blend state of Output Merger
    (see D3D11_BLEND_DESC, Hint: $F \leftrightarrow SRC, B \leftrightarrow DST$)

# Example

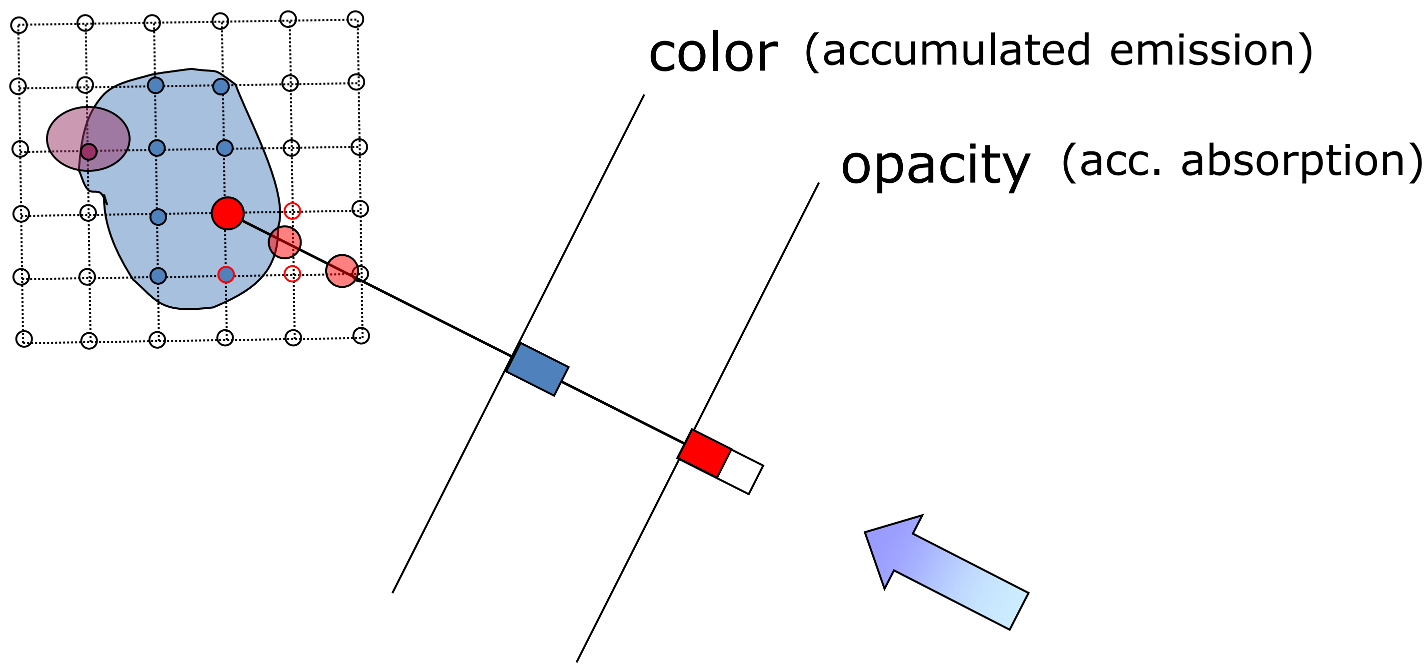


color (accumulated emission)
opacity (acc. absorption)
Trilinear Interpolation!
1.0

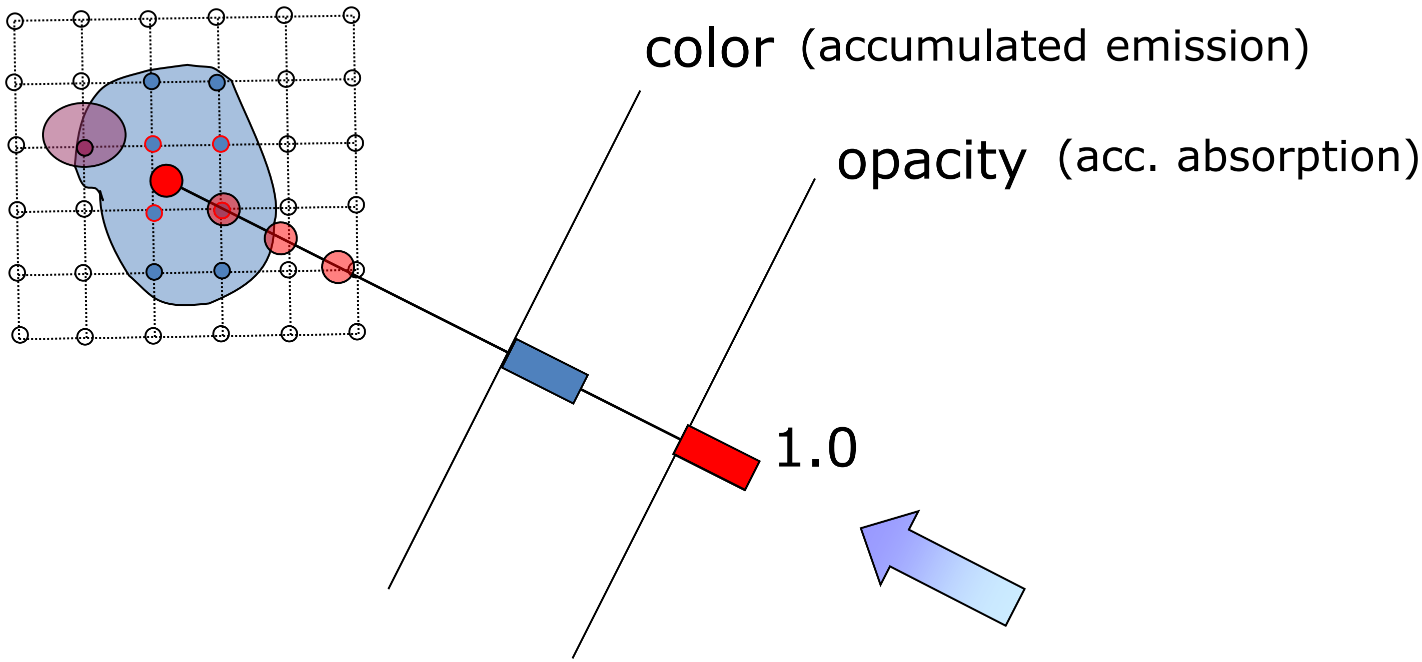
# Example



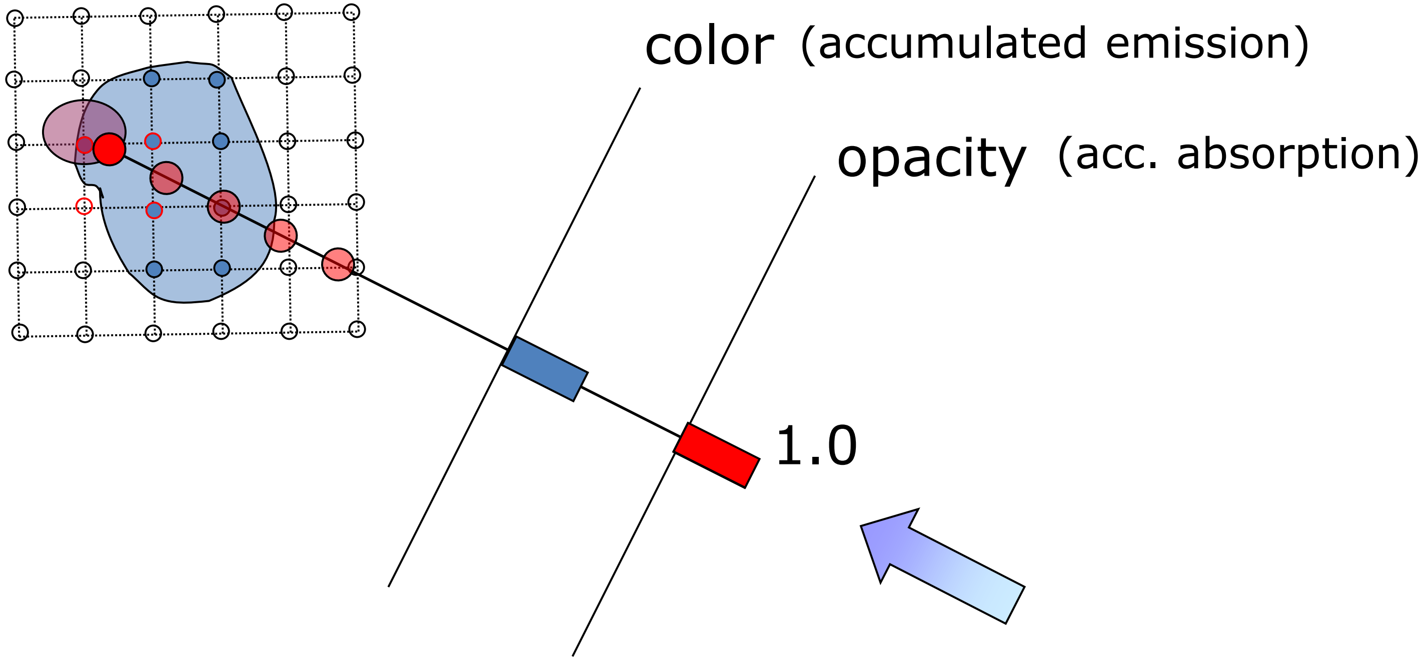
# Example



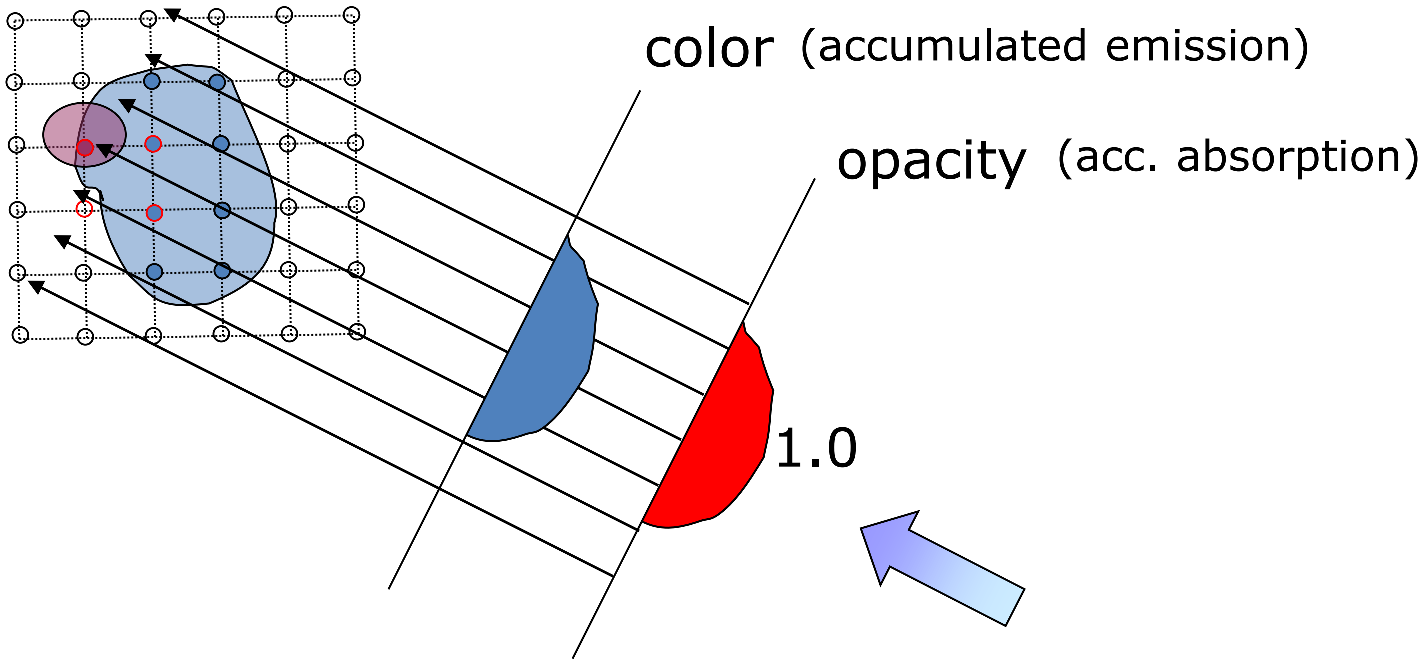
# Example



# Example



# Example



# Q&A

computer graphics & visualization

**Questions ?**