VMV 2016

Real-time Novel-view Synthesis for Volume Rendering Using a Piecewise-analytic Representation

Gerrit Lochmann [1] Bernhard Reinert [2] Arend Buchacher [1] Tobias Ritschel [2,3]

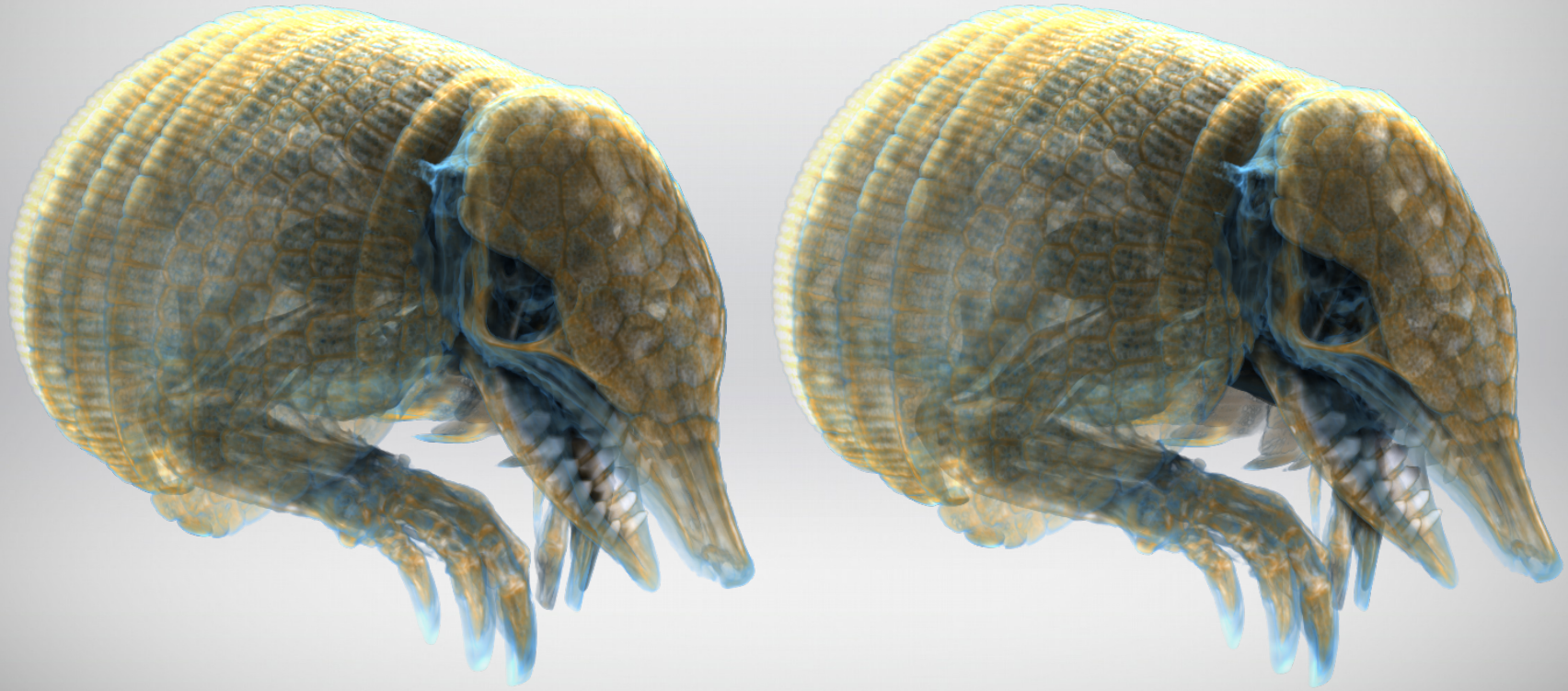
UNIVERSITÄT KOBLENZ · LANDAU [1]

max planck institut informatik [2]

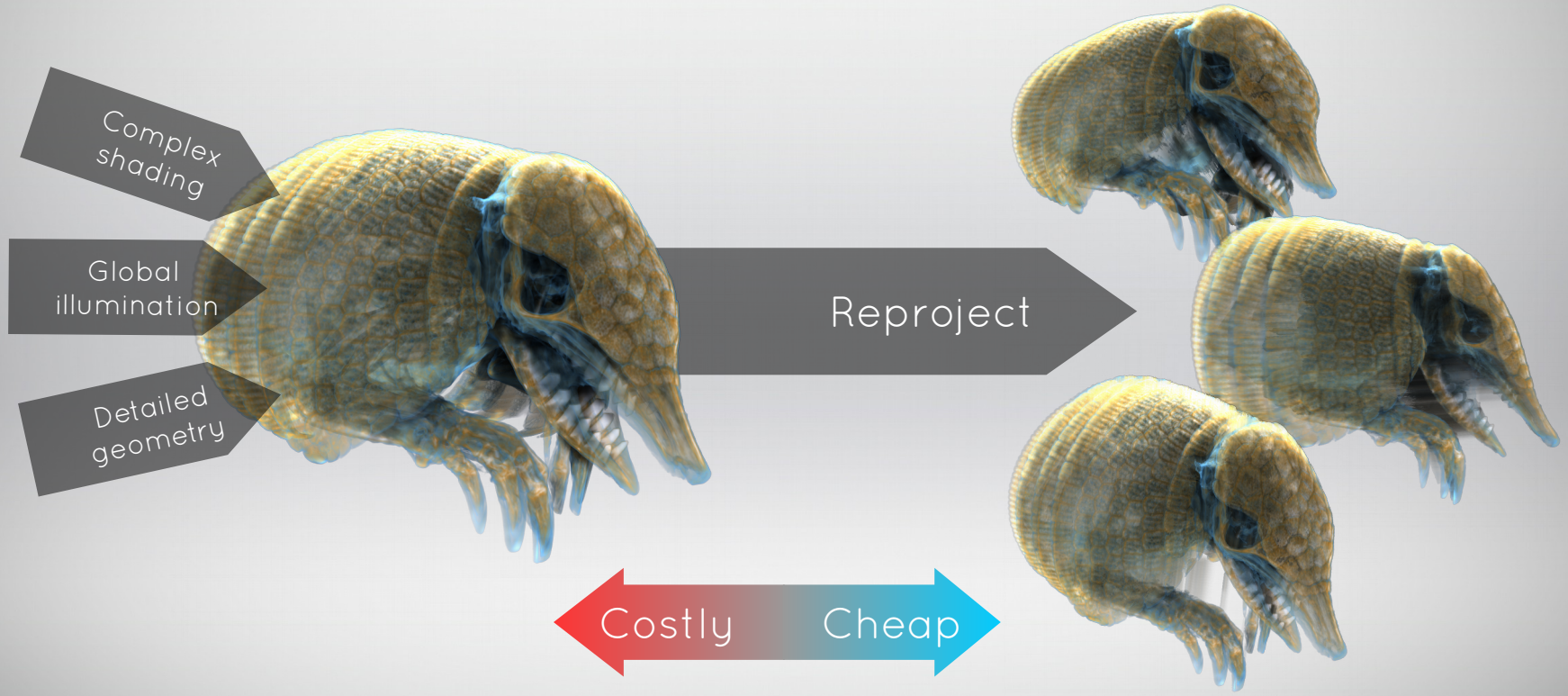
UCL [3]

Motivation

Coherency between images

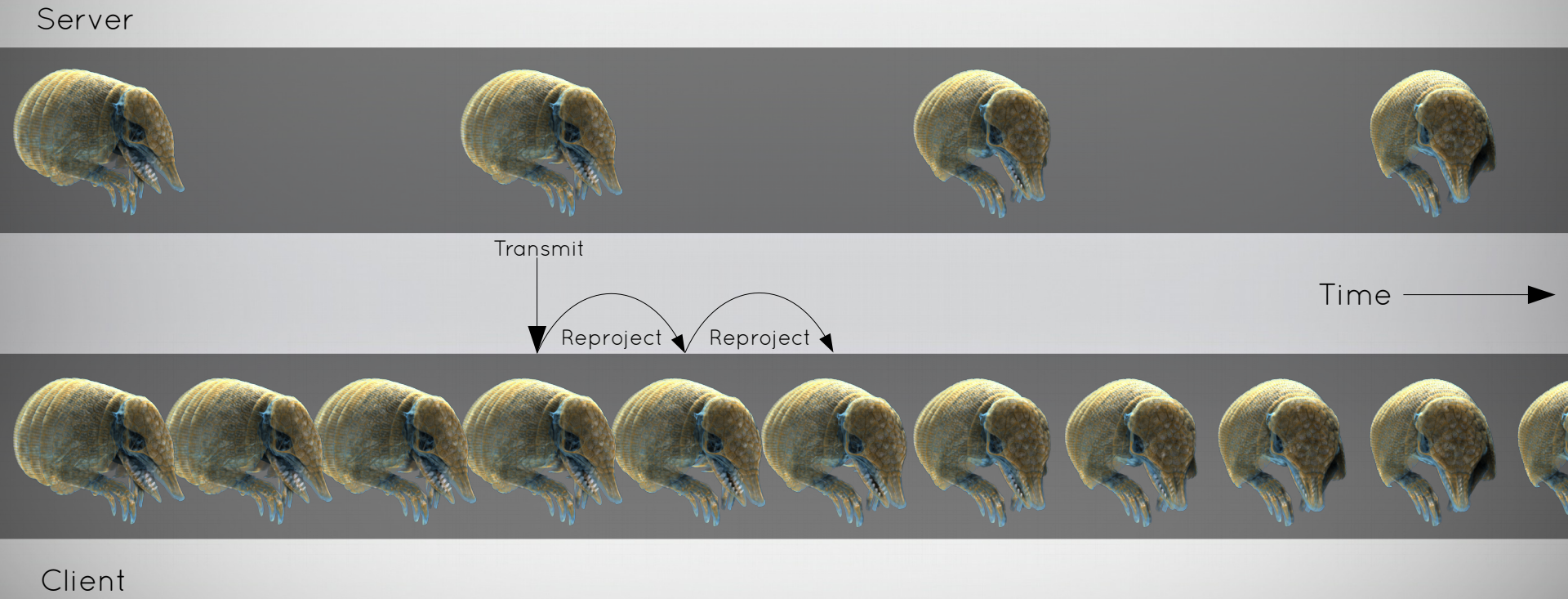


Avoid costly computations for slightly different views

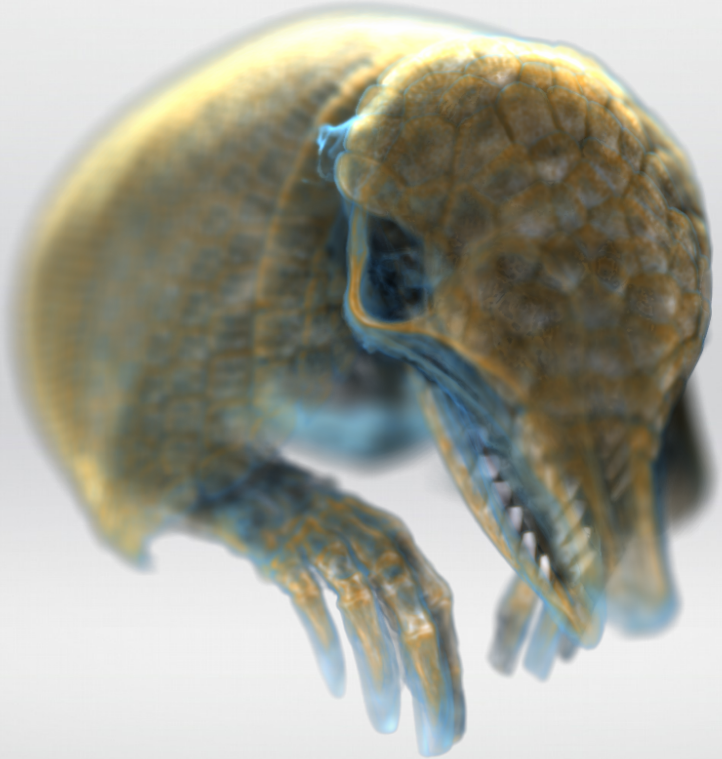


Application: Stereo synthesis

Application: In-between-frames

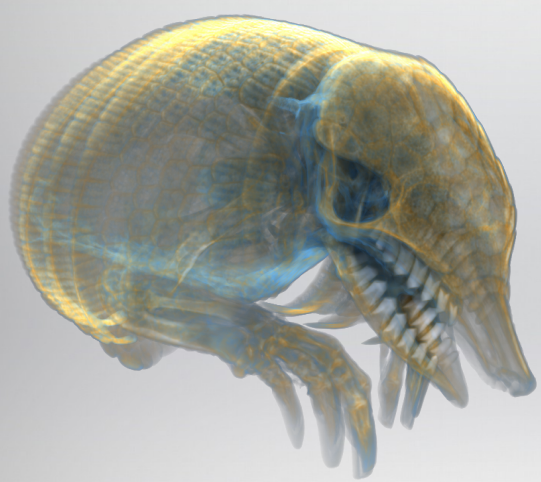


Application: Distribution effects / depth-of-field

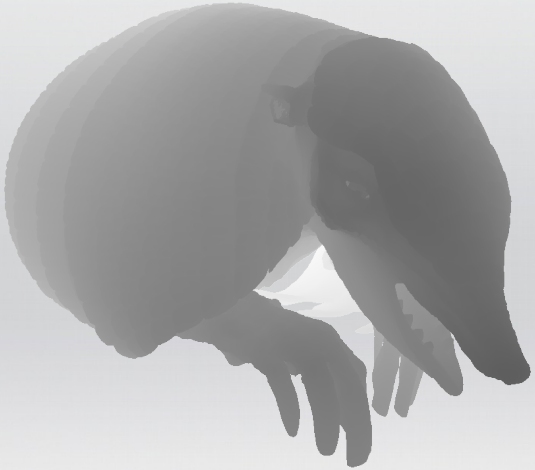


Related work

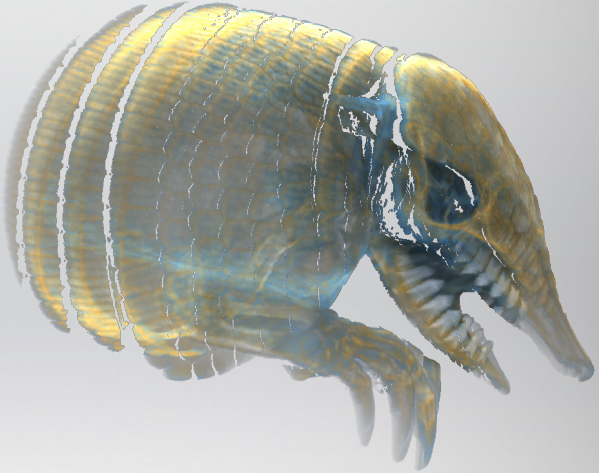
Post-Rendering 3D Warping [Mark et al. 1997]



Color image



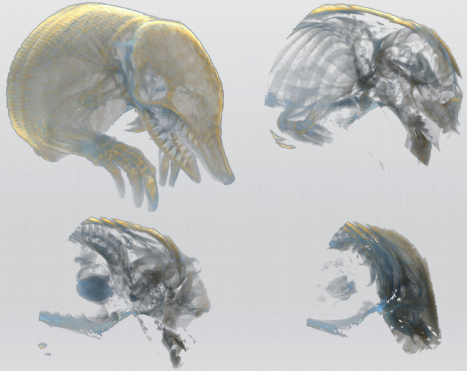
Depth information



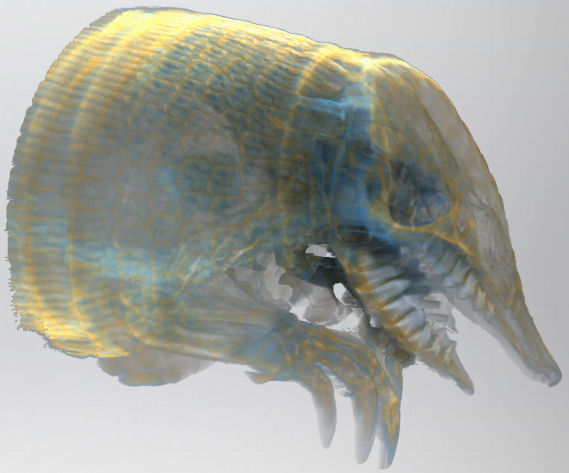
Move pixels

IBR-Assisted Volume Rendering [Müller et al. 1999]

Depth layers

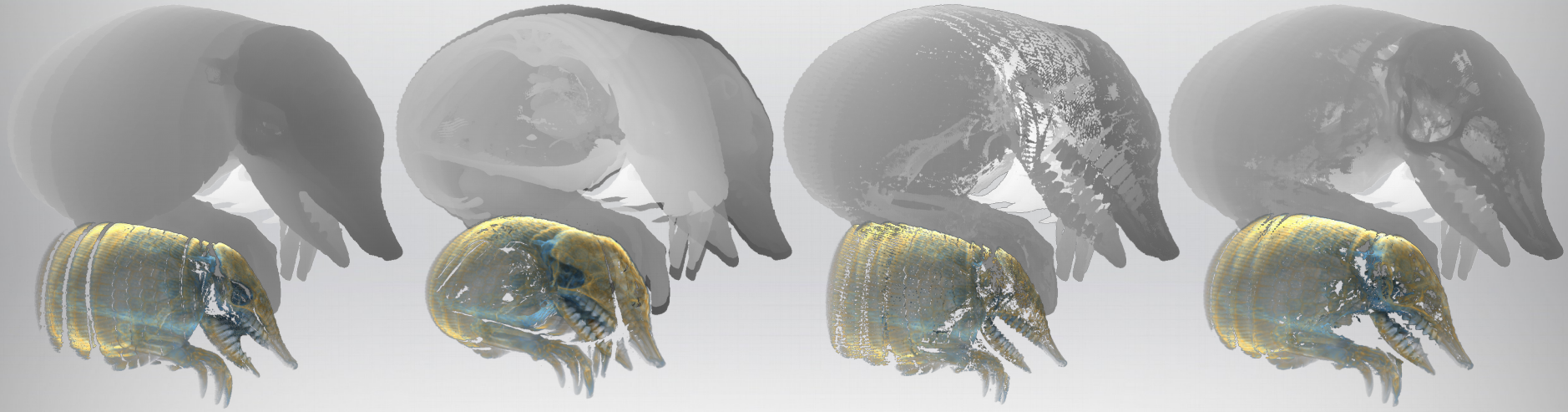


Piecewise integration



Layerwise Reprojection and Compositing

Image-Based Remote Real-Time Volume Rendering [Zellmann et al. 2013]



Entry point

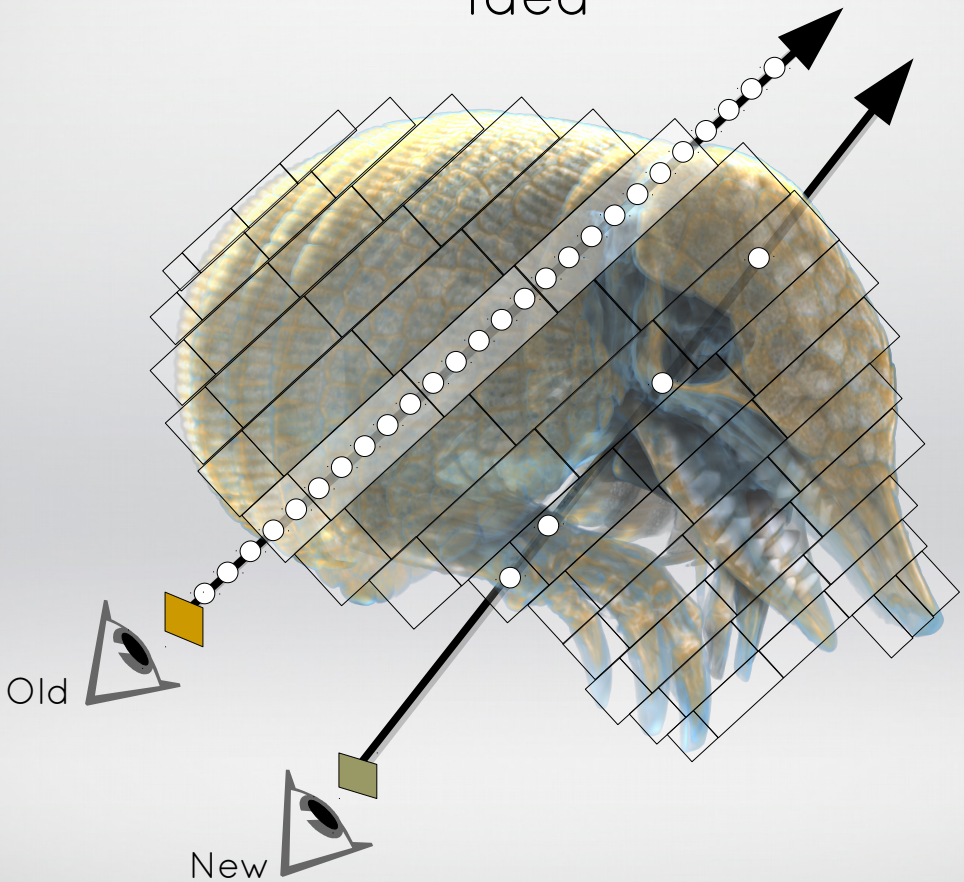
Entry-exit midpoint

Highest density

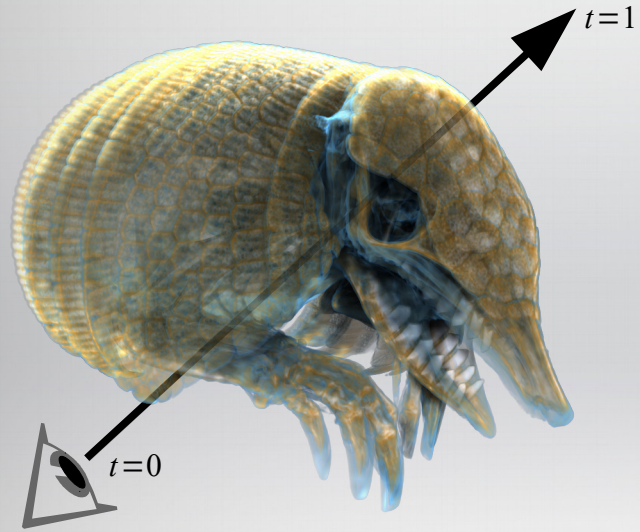
Accumulated density threshold

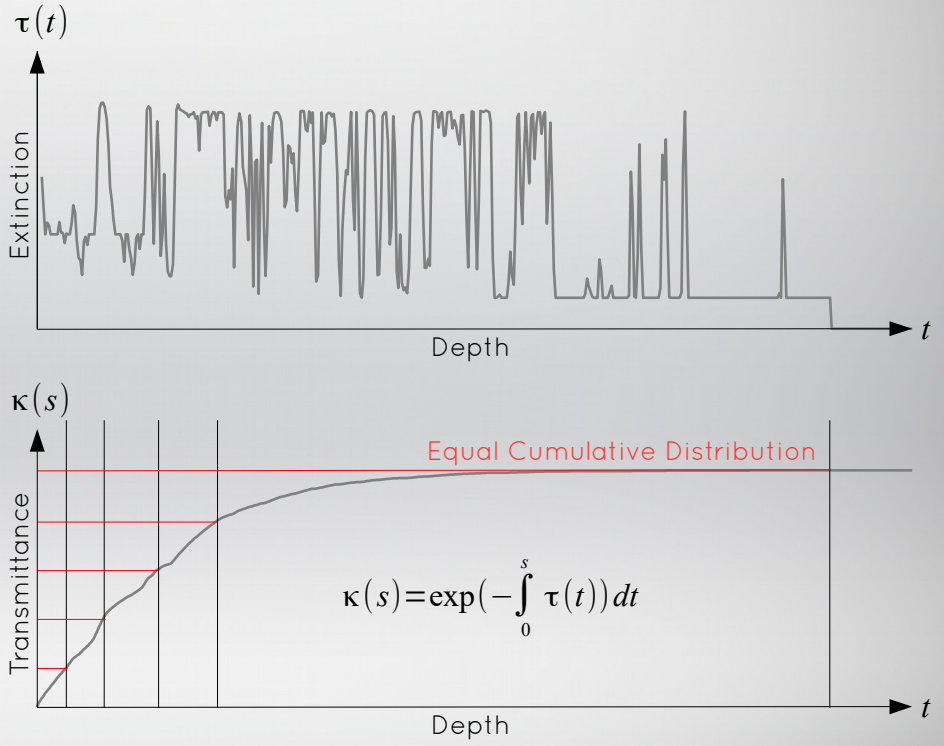
Our Algorithm

Idea

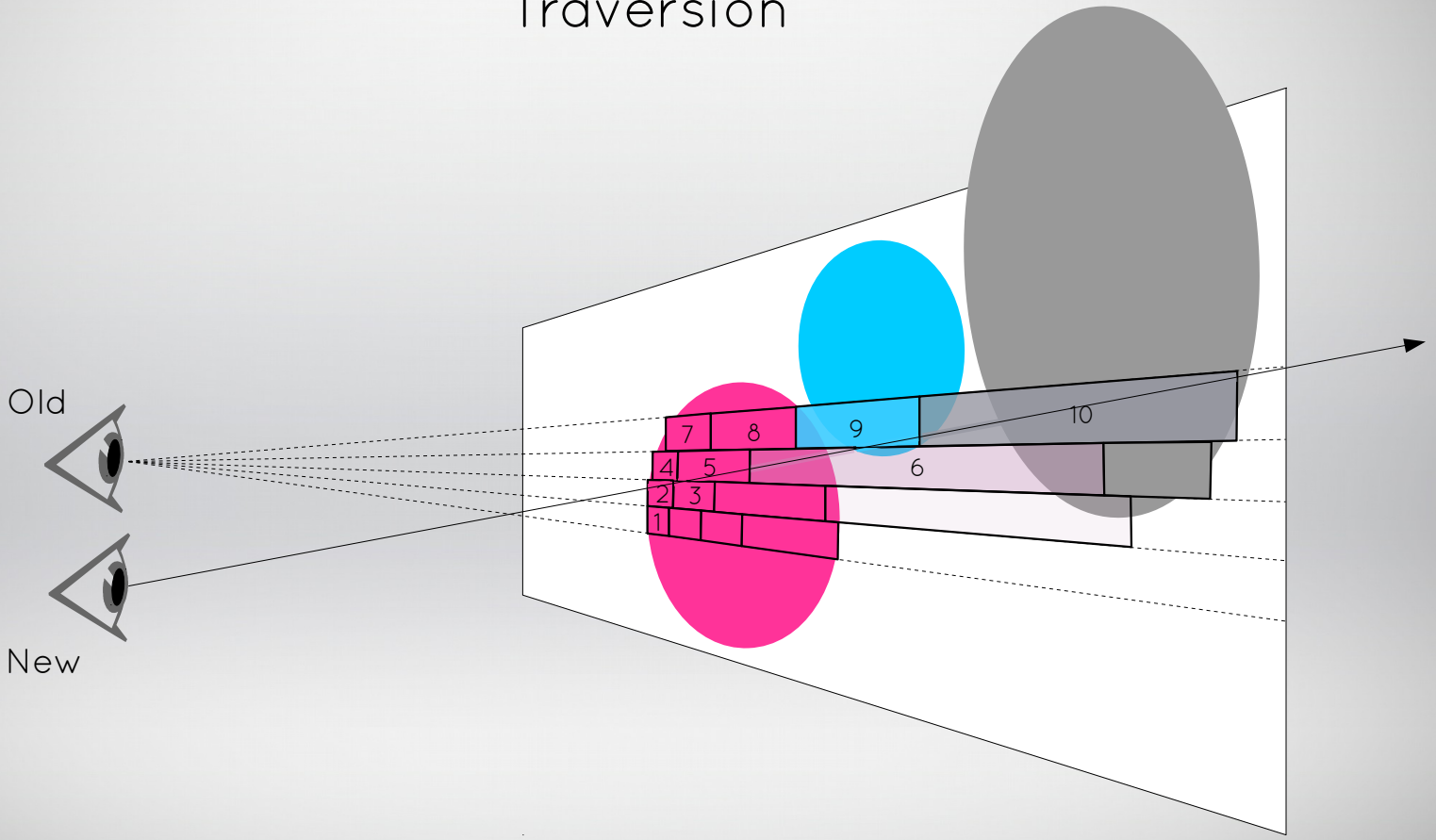


Dividing the ray



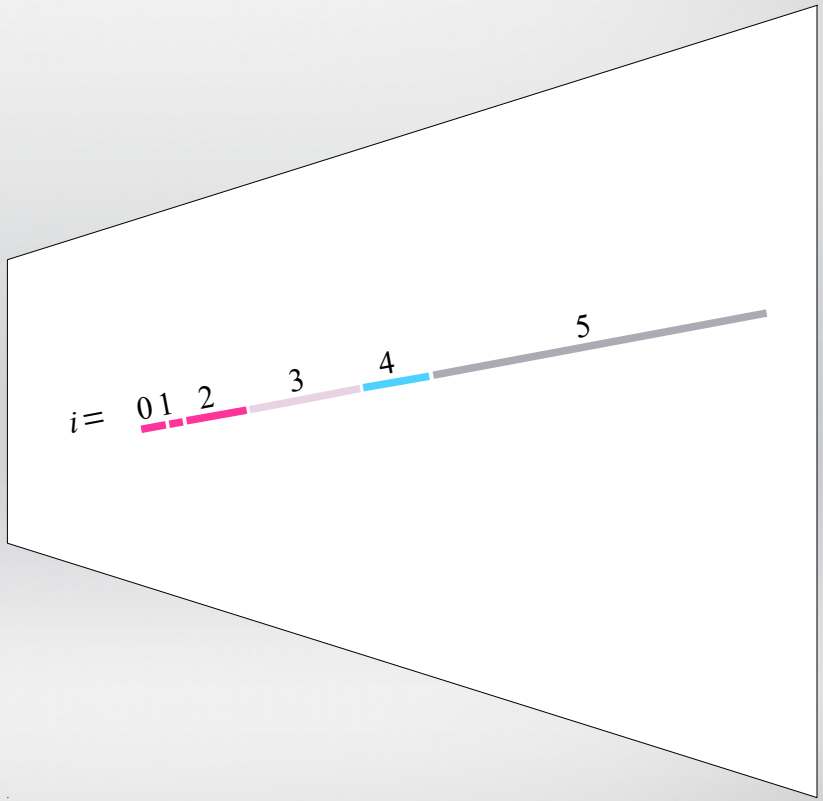


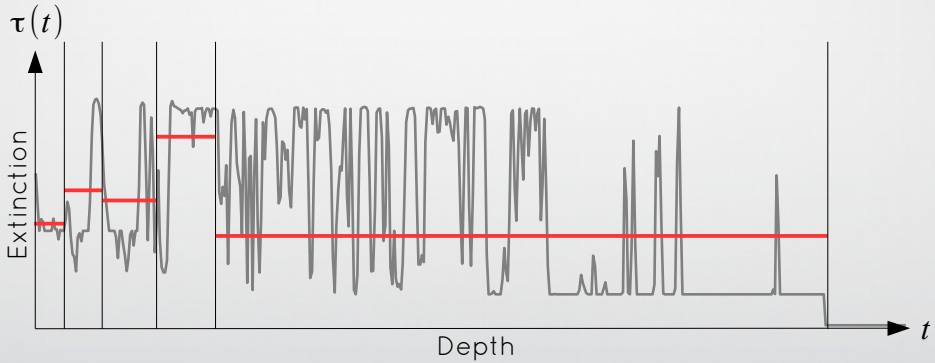
Traversion



Traversion

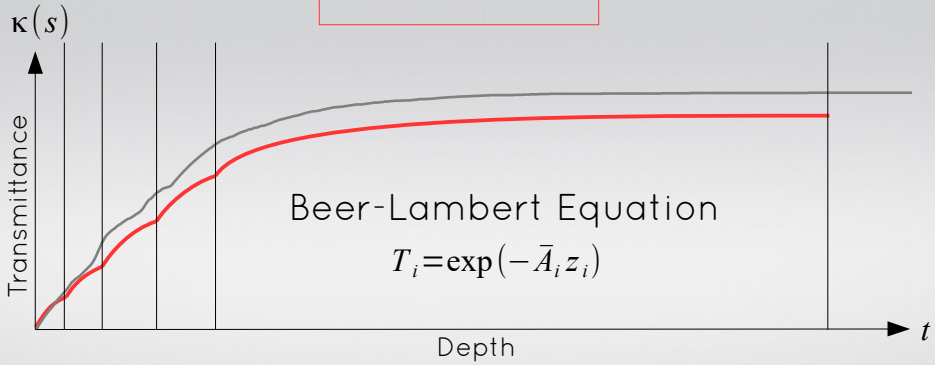
New



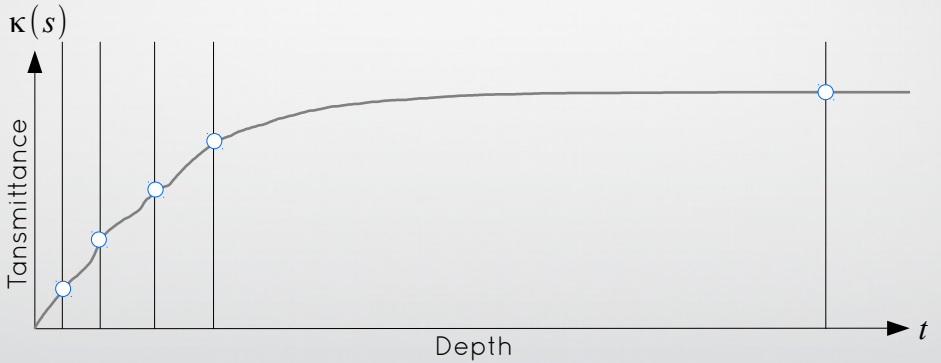


Encoding

$$\bar{A}_i = \frac{1}{z_i} \int_{s_i}^{s_{i+1}} \tau(t)\, ds$$

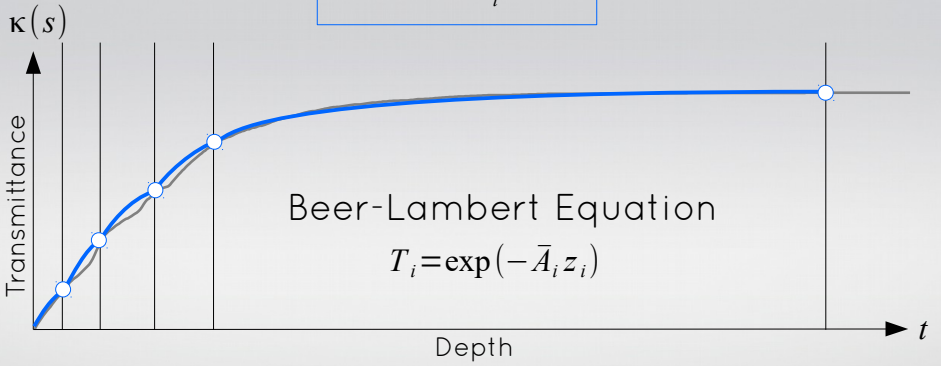


Decoding
from original view



Encoding

$$A_i = \frac{-\log(T_i)}{z_i}$$



Decoding
from original view

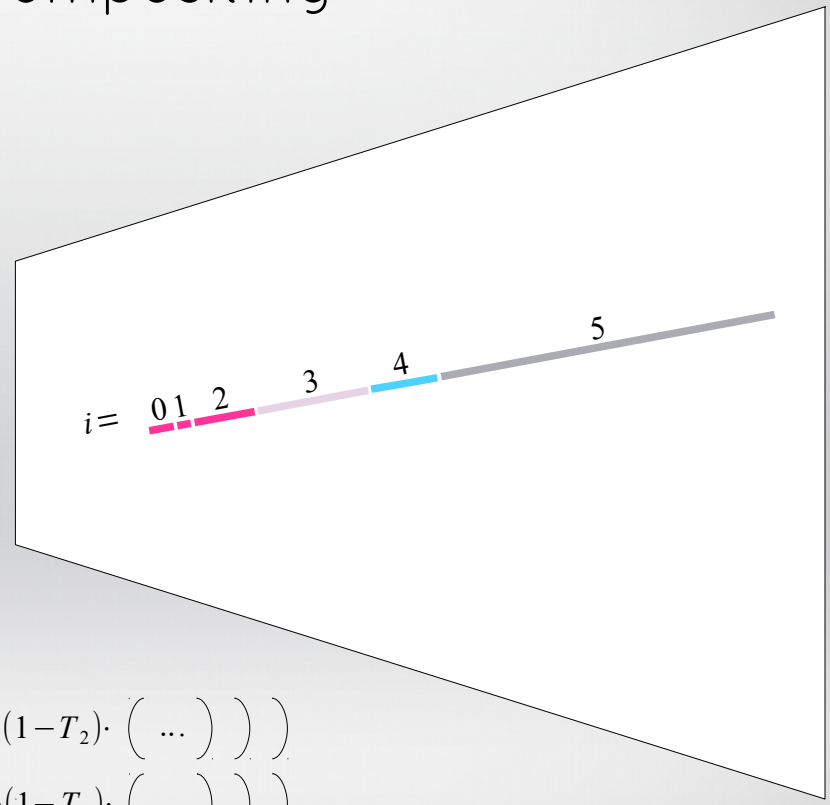
Compositing

Opacity per segment

$$T_i = \exp(-A_i z_i)$$

Color per segment

$$C_i = E_i \cdot T_i$$



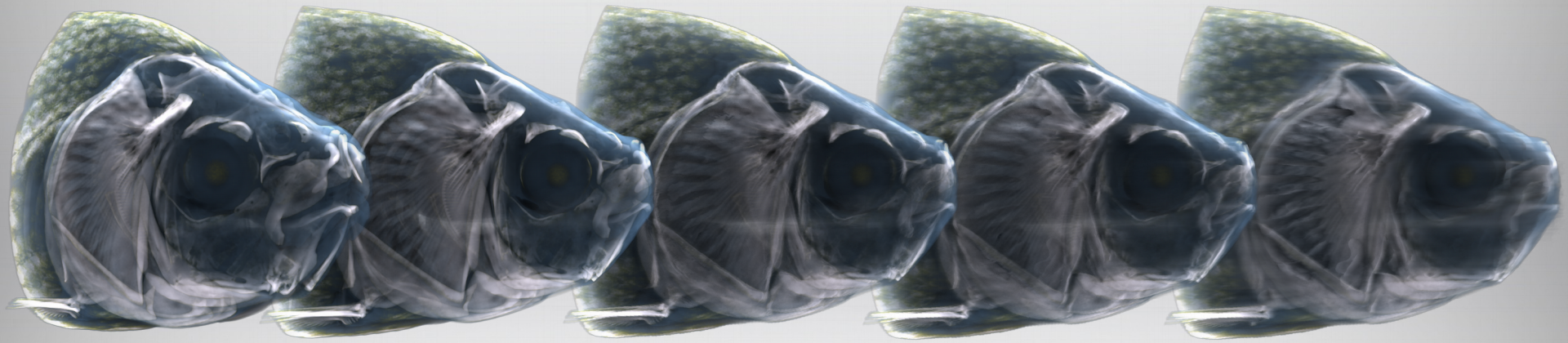
Front-to-back compositing

$$\kappa(x) = T_0 + (1-T_0)\cdot\left(T_1 + (1-T_1)\cdot\left(T_2 + (1-T_2)\cdot\left(\ldots\right)\right)\right)$$

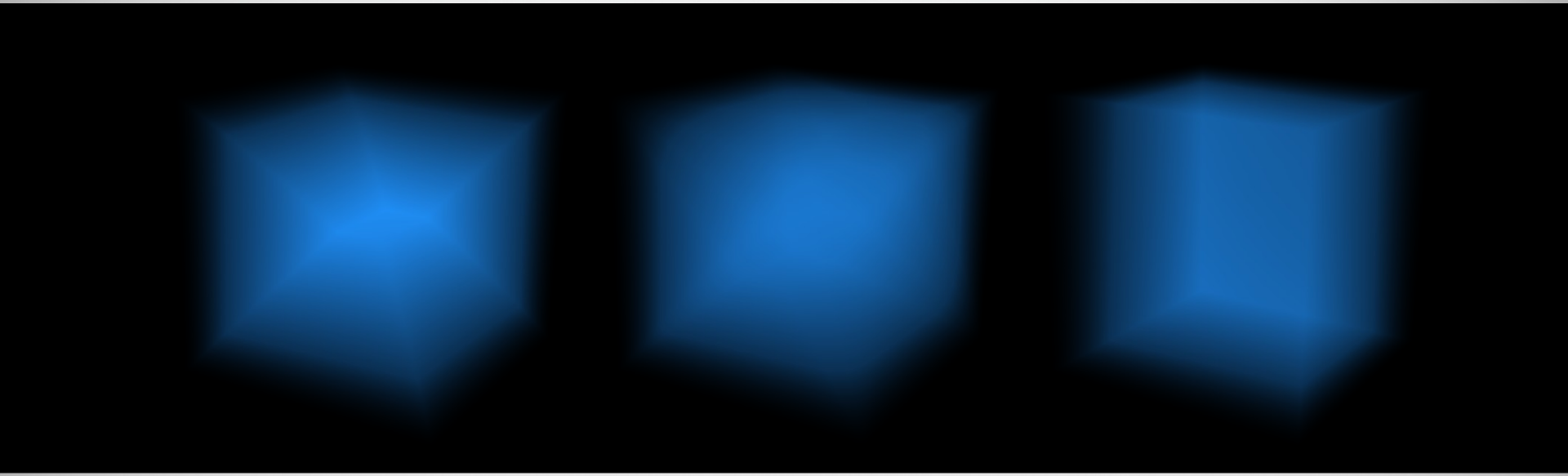
$$\eta(x) = C_0 + (1-T_0)\cdot\left(C_1 + (1-T_1)\cdot\left(C_2 + (1-T_2)\cdot\left(\ldots\right)\right)\right)$$

Results

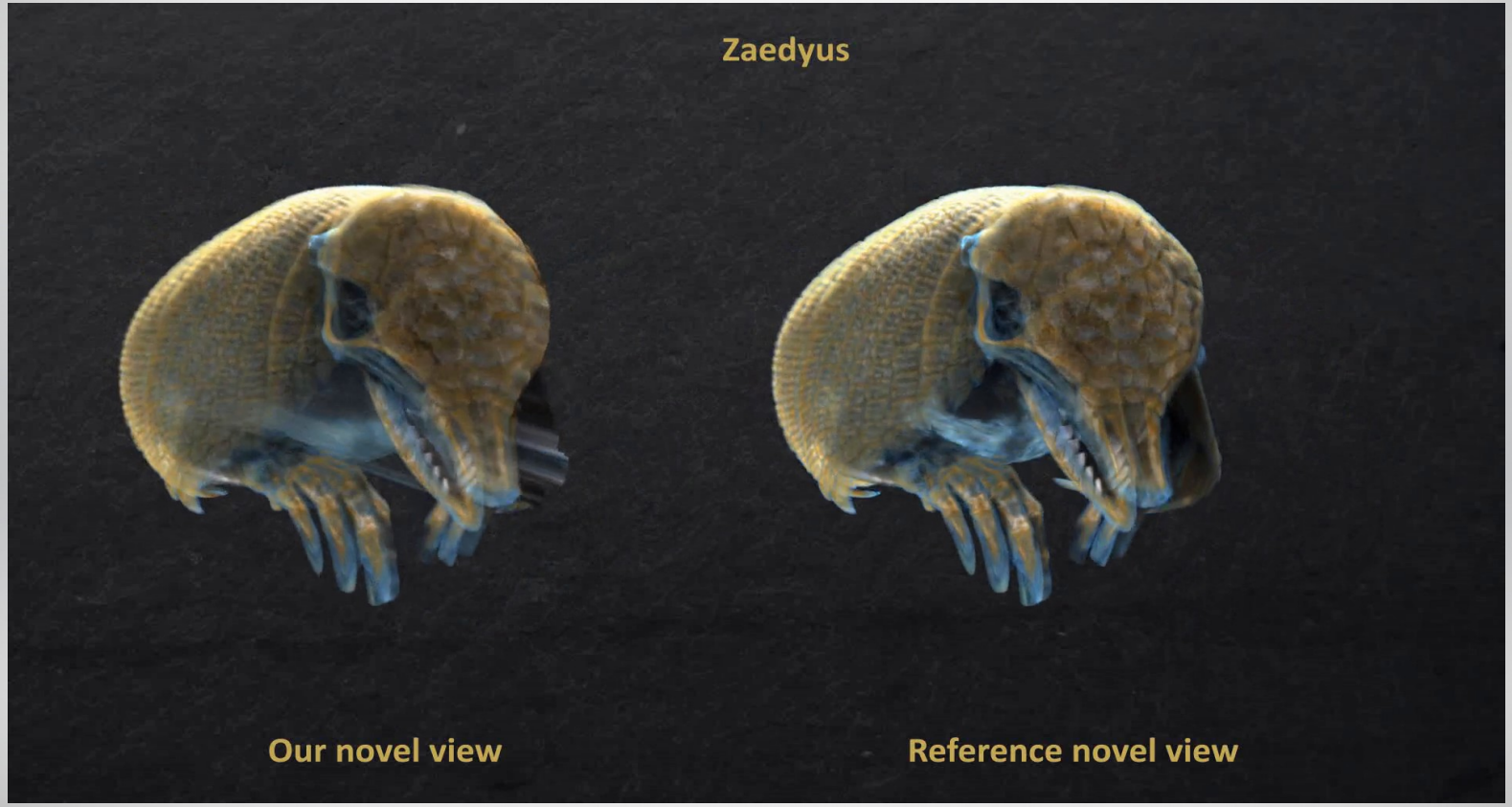
Visual Quality depending on the layer count

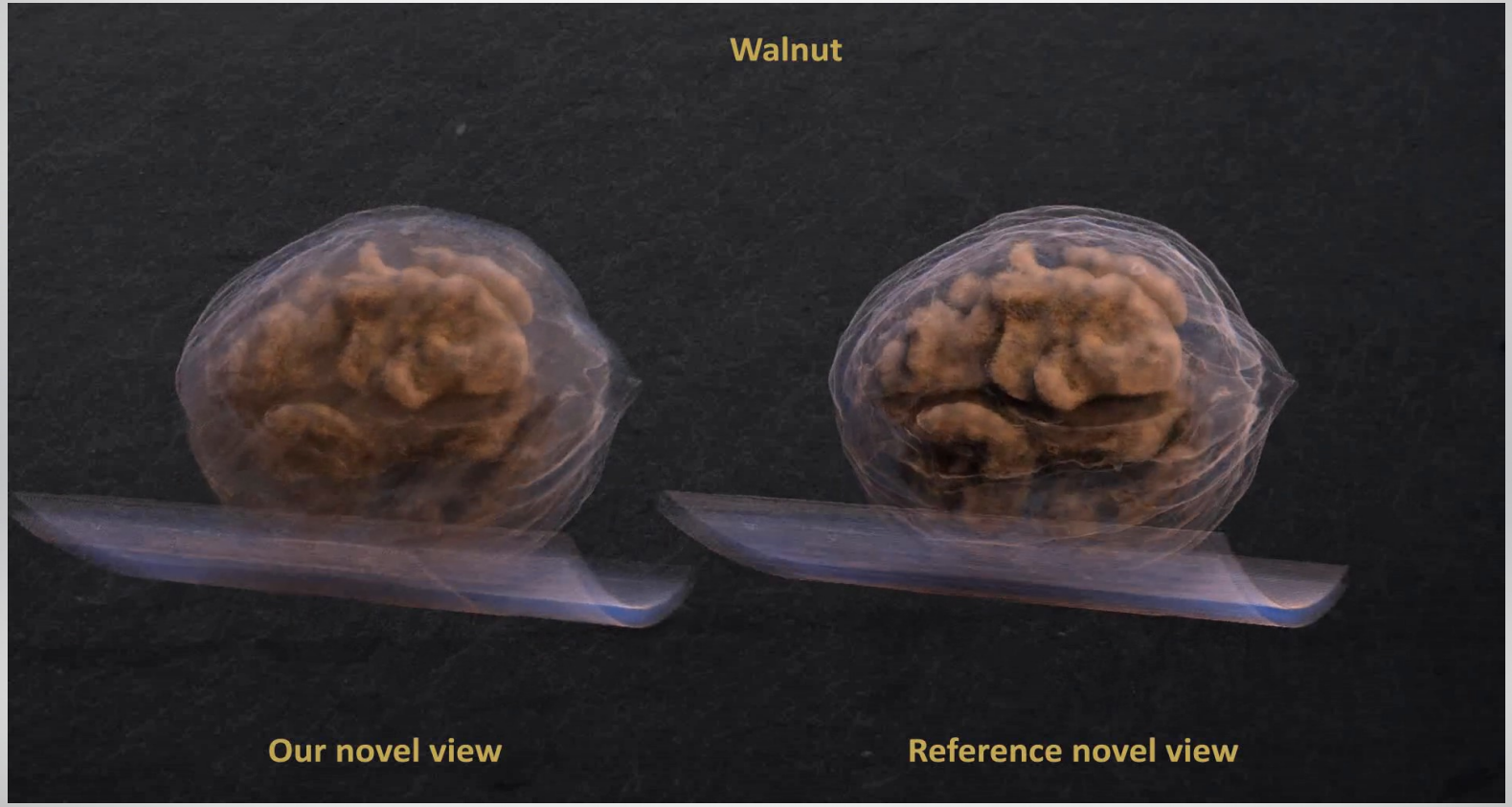


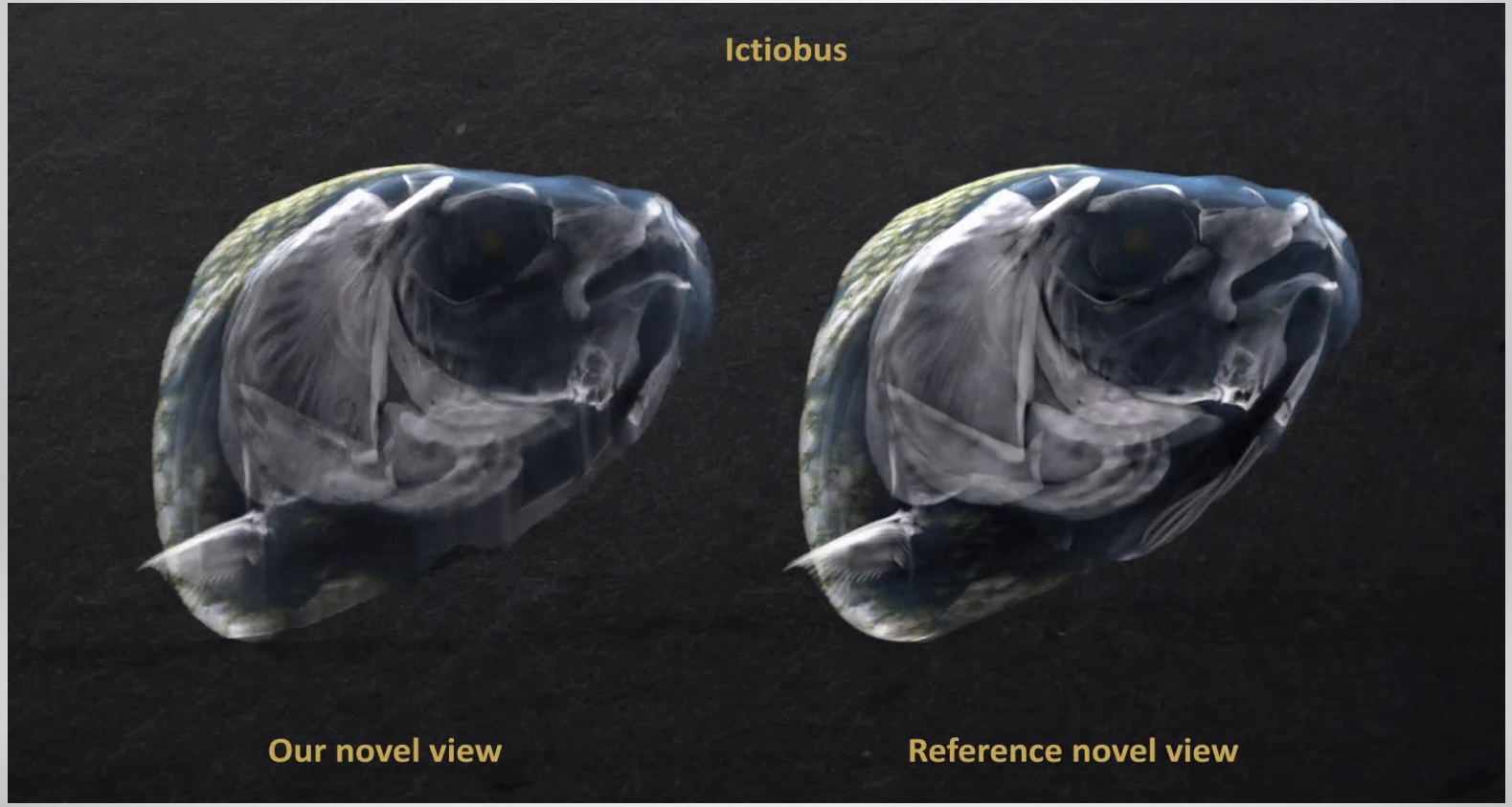
Source | Reference | 8 Layers | 4 Layers | 2 Layers

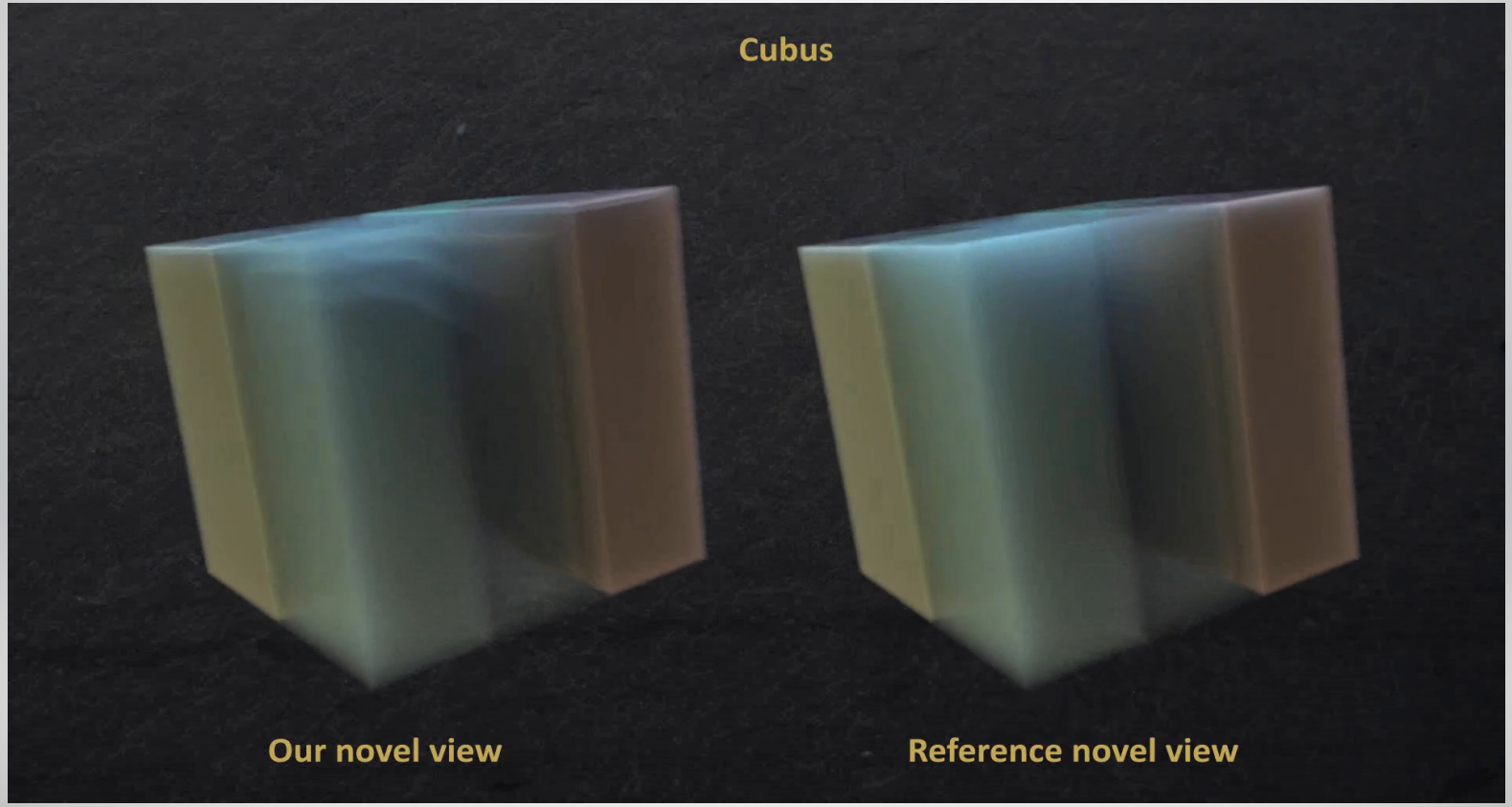


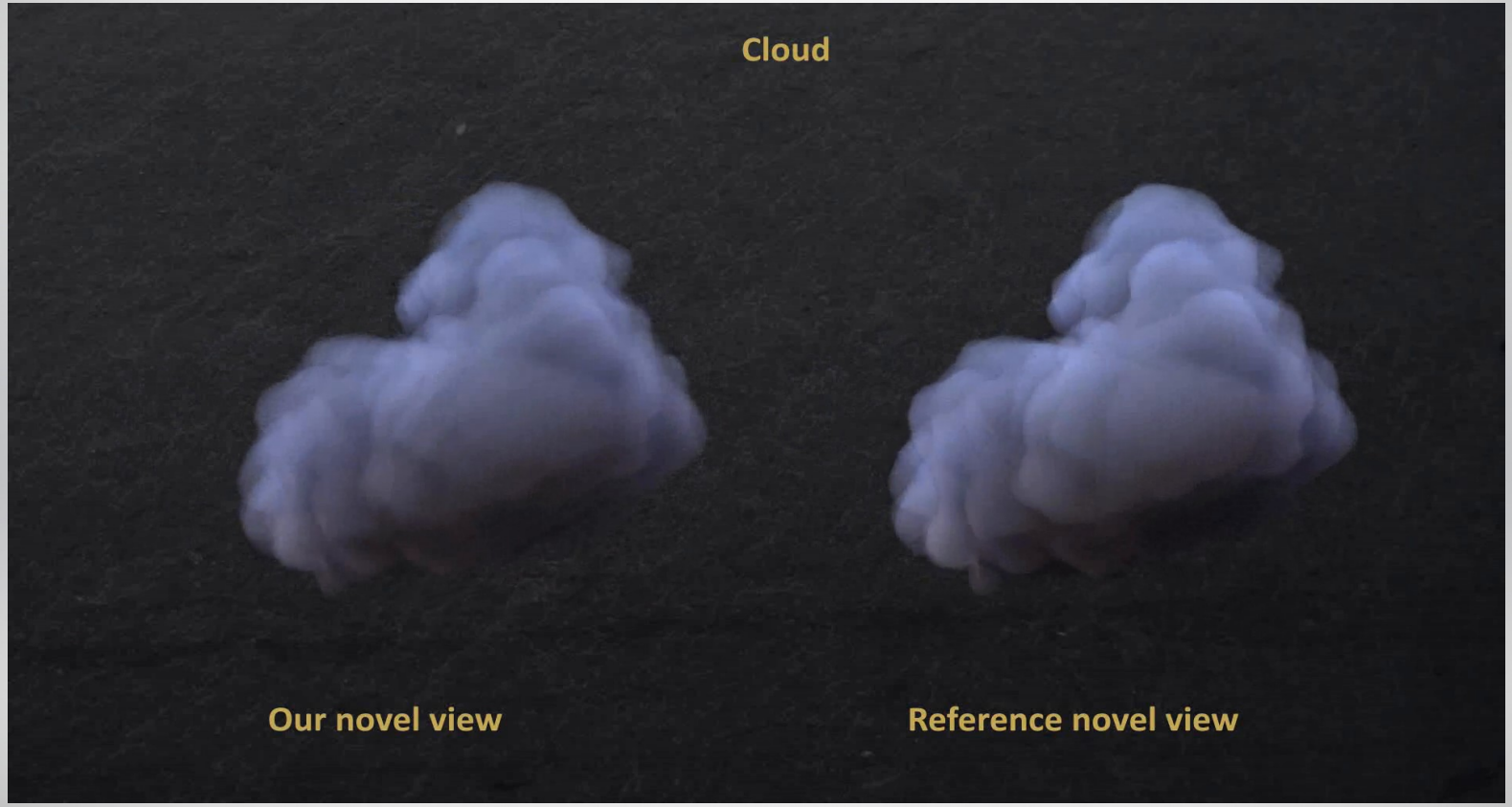
Zellmann et al. (2013) Müller et al. (1999) Ours

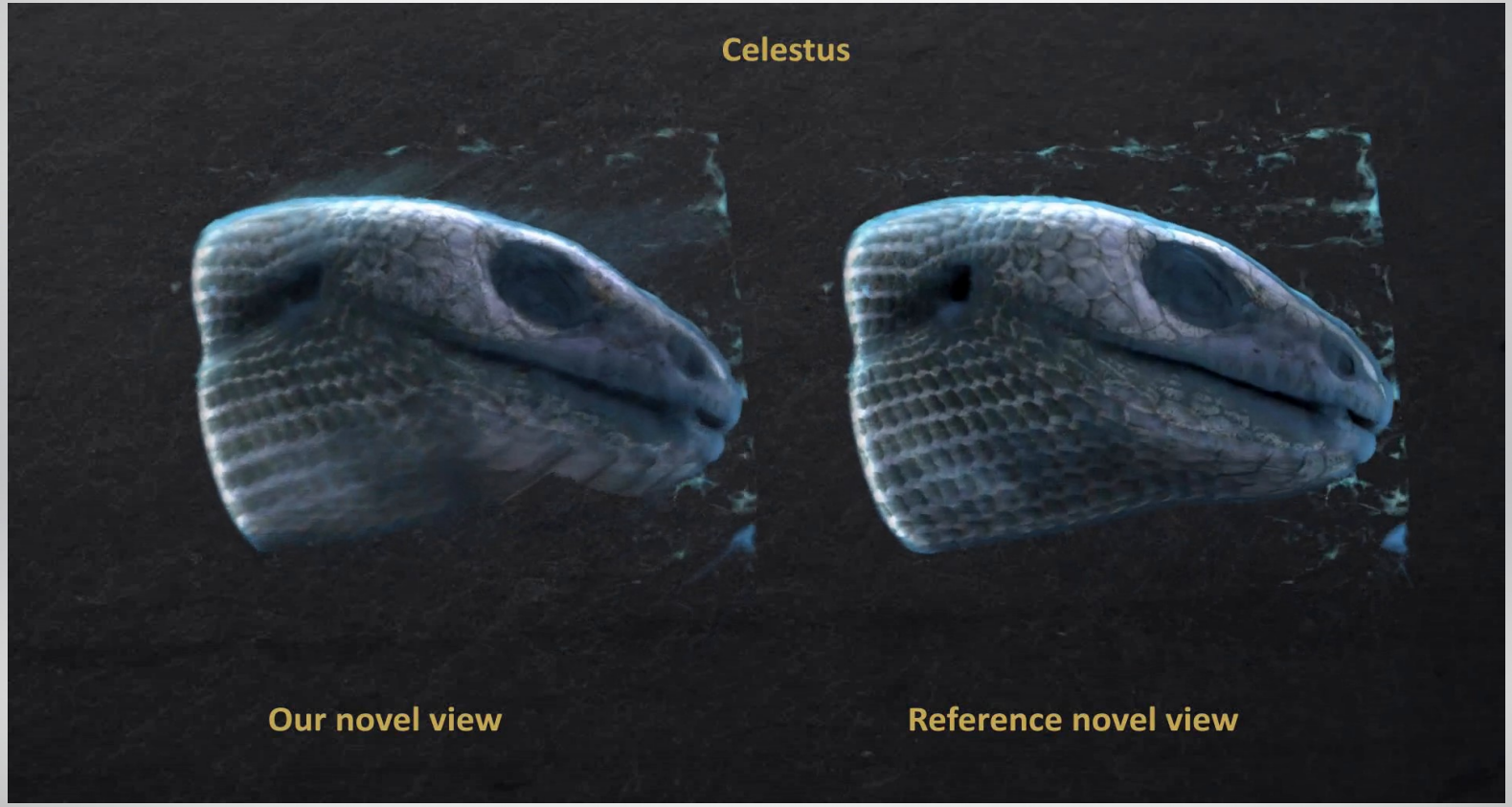

Zaedyus
Our novel view
Reference novel view


Walnut
Our novel view
Reference novel view

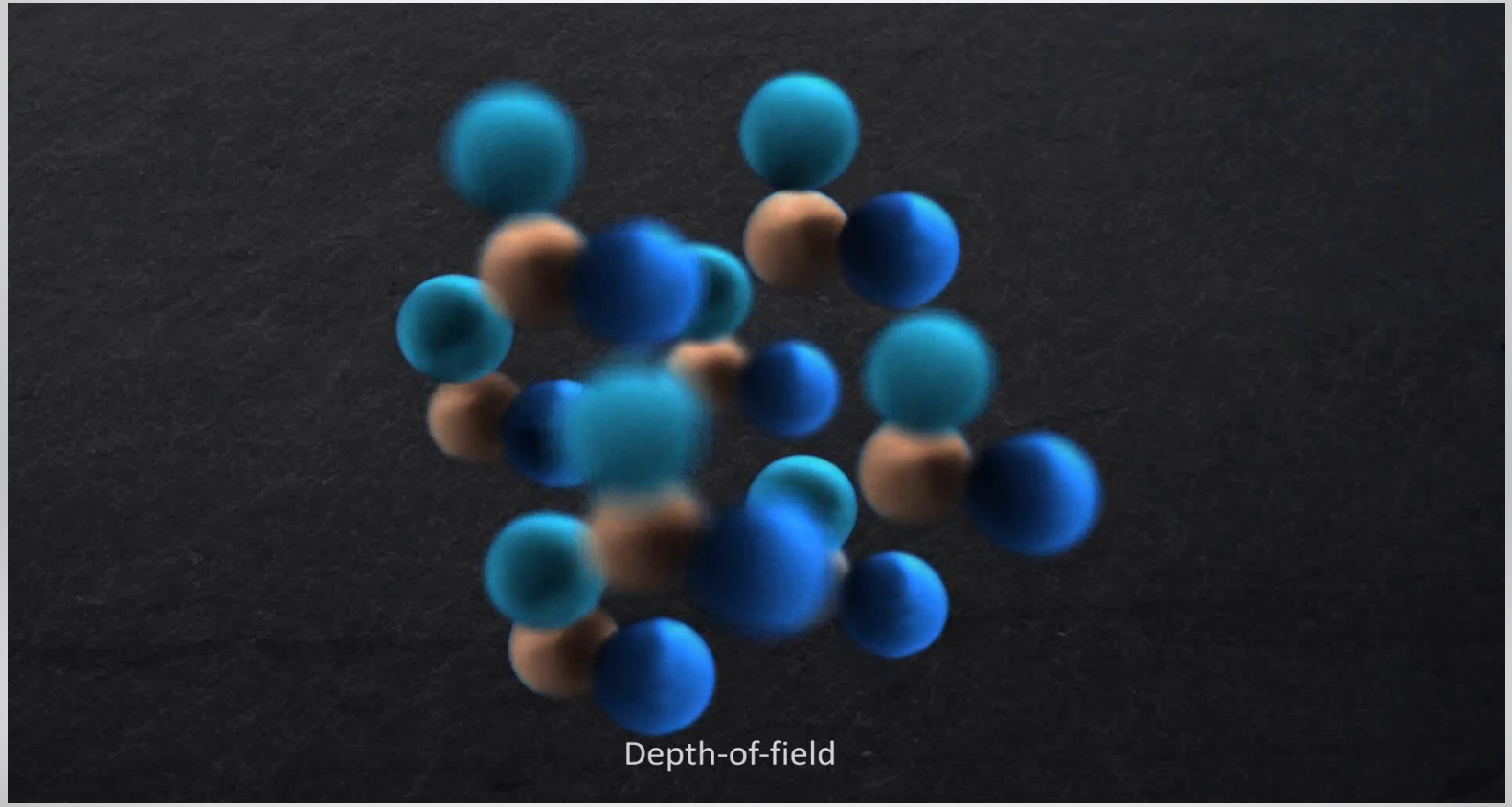

Ictiobus
Our novel view
Reference novel view


Cubus
Our novel view
Reference novel view


Cloud
Our novel view
Reference novel view


Celestus
Our novel view
Reference novel view

Applications


Depth-of-field


Latency-compensated volume rendering

Conclusion

Benefits

- Novel views in **real-time without reshading**
- **Original color** from the original view
- **More accurate** results than state-of-the-art competitors

Limitations

- Missing details → **stretching artifacts**
- **Cost** and **bandwidth** increase with accuracy (layer count)
- We assume an isotopic phase function
  → **view dependent shading** is not supported

Thank you!

Appendix

Error (DSSIM) among competitors

	Ours	Müller	Zellmann
Zaedyus	.031	.041	.081
Cubus	.005	.006	.010
Ictiobus	.027	.022	.070
Celestus	.036	.070	.067
Cloud	.013	.036	.064
Walnut	.025	.063	.109

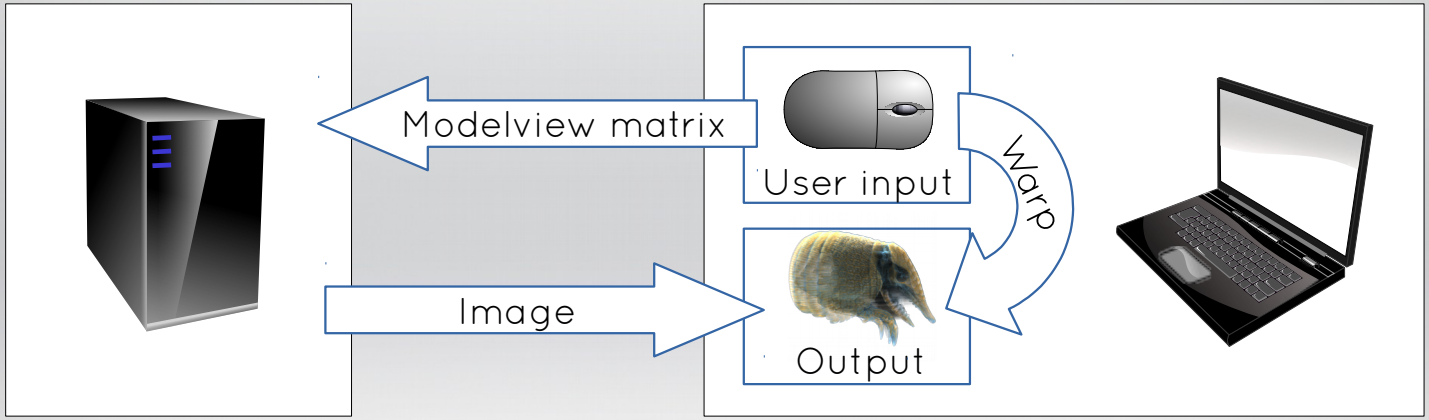
10° view rotation, 1024 x 1024 pixels

Error (DSSIM) and speed (in ms)

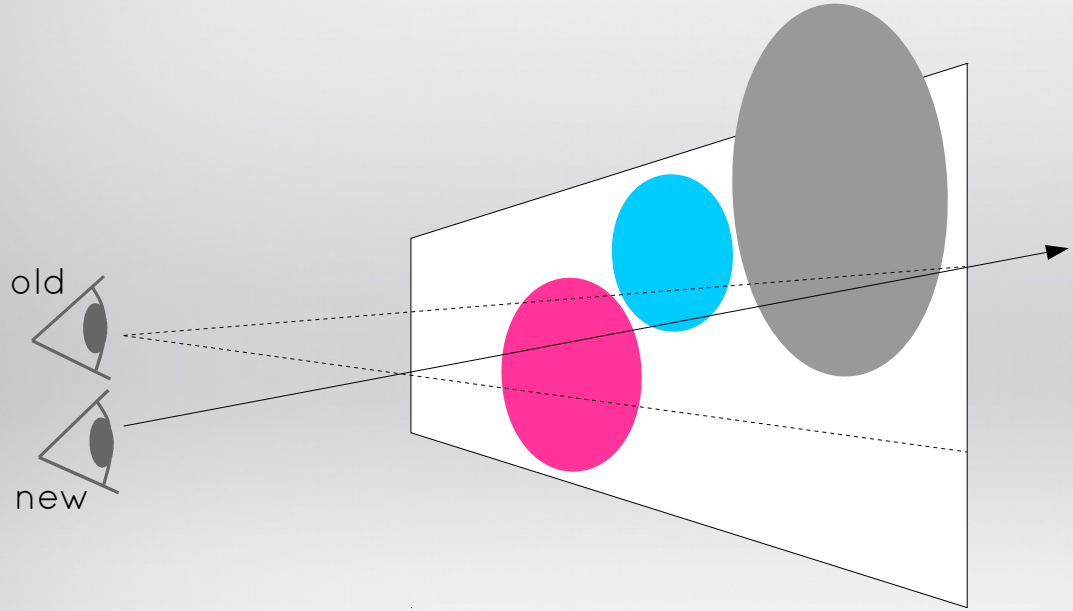
	2 Layers		4 Layers		8 Layers	
	Error	Speed	Error	Speed	Error	Speed
5 Degrees	.035	22.7	.024	24.3	.016	29.0
10 Degrees	.049	24.5	.031	27.6	.020	35.6
15 Degrees	.060	27.7	.037	30.9	.023	41.8

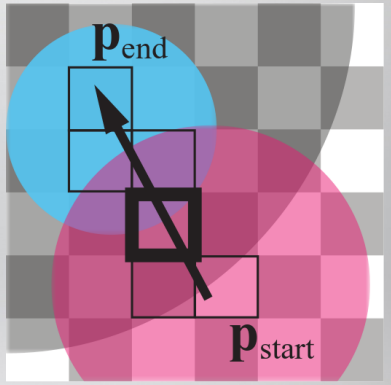
View rotation, Zaedyus data set, 1024 x 1024 pixels, GeForce GTX 980

Application: Remote rendering / latency reduction



Client





DDA in the old view