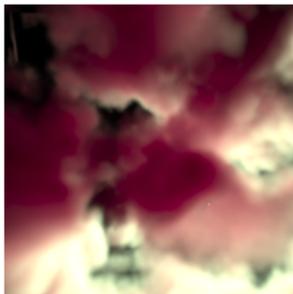


Efficient spectral volumetric rendering based on astrophysical assumptions

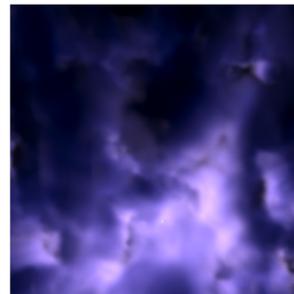


Julien Barrès

Supervisor: Fabrice Neyret

Maverick team: INRIA - LJK

June 26, 2023



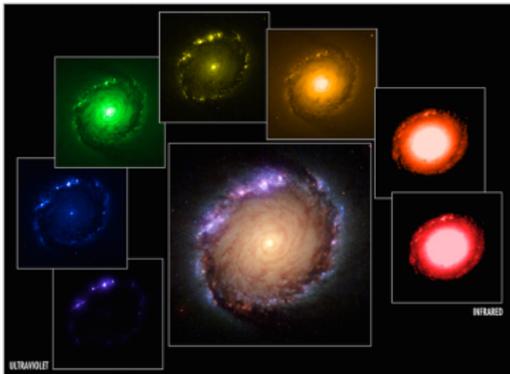
Motivation



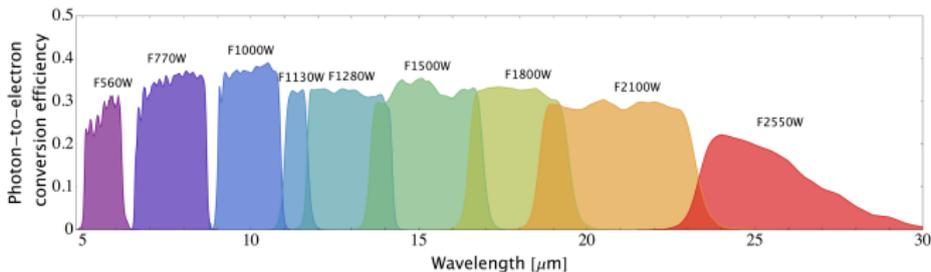
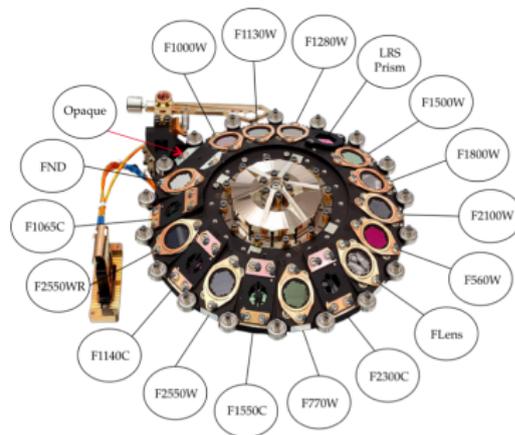
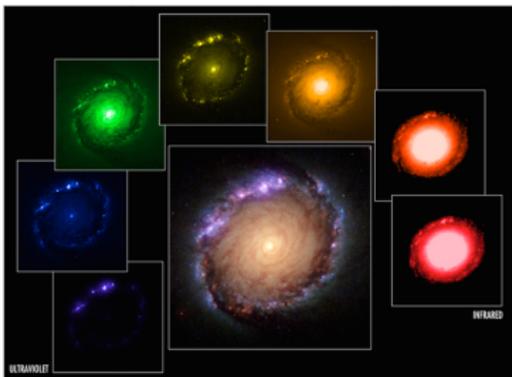
- Rendu de scènes astronomiques
- Temps réel
- Réaliste, physique

¹INRIA ANR Galaxy/veRTIGE.

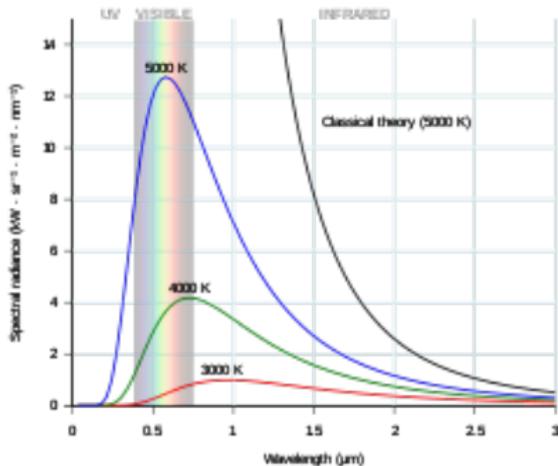
Astrophotography imagery process



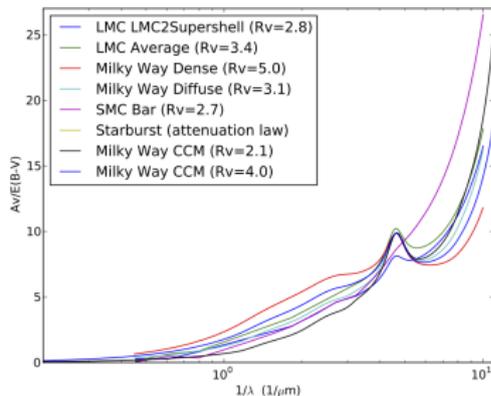
Astrophotography imagery process



Other involved spectra



Emission spectrum



Absorption spectrum

Nebulae

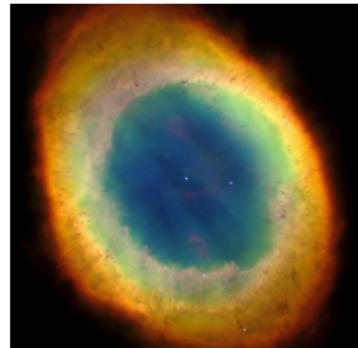
Nebulae are complex objects interacting with the light passing through them (**participating medium**).



Absorption nebula

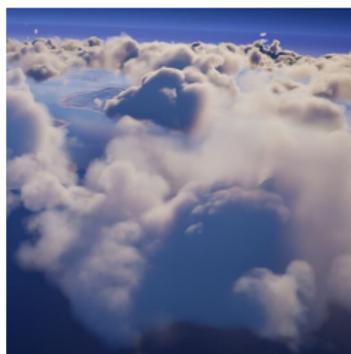
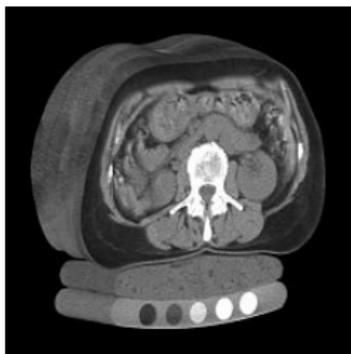


Reflection nebula

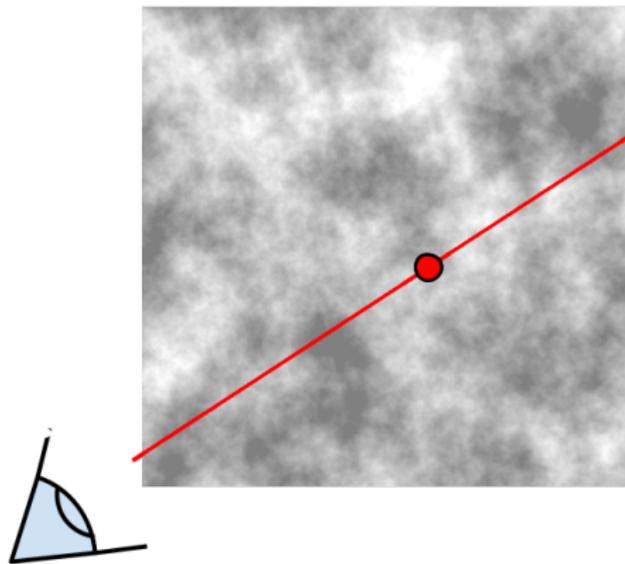


Emission nebula

Spectral volumetric rendering



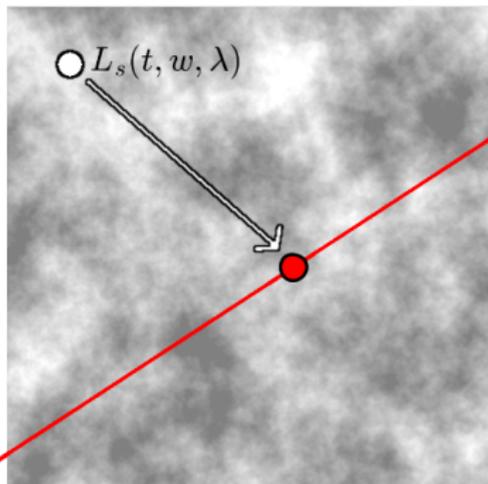
Volumetric rendering equation²



$$L_{local,\lambda}(t, w) = [\quad]$$

²Kajiya and Herzen, "Ray Tracing Volume Densities".

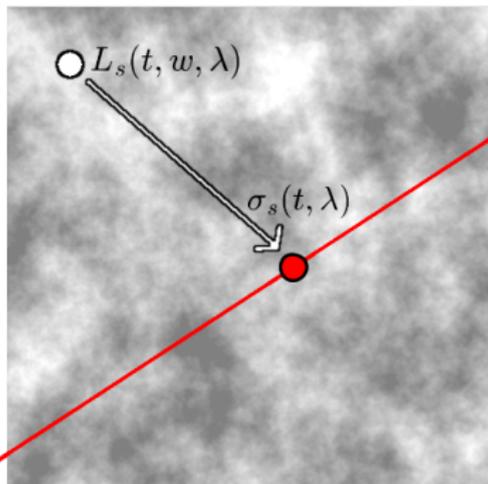
Volumetric rendering equation²



$$L_{local,\lambda}(t, w) = \int [L_s(t, w, \lambda) \dots]$$

²Kajiya and Herzen, "Ray Tracing Volume Densities".

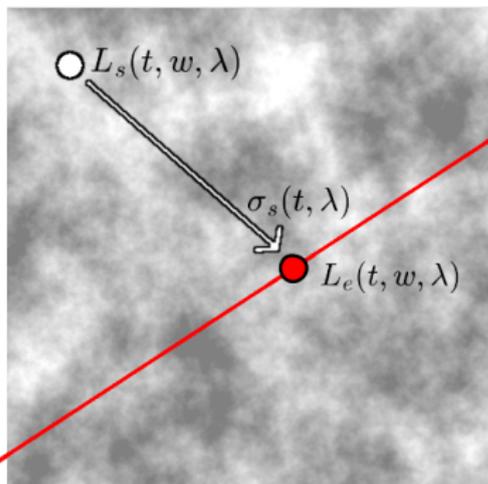
Volumetric rendering equation²



$$L_{local,\lambda}(t, w) = [L_s(t, w, \lambda)\sigma_s(t, \lambda)]$$

²Kajiya and Herzen, "Ray Tracing Volume Densities".

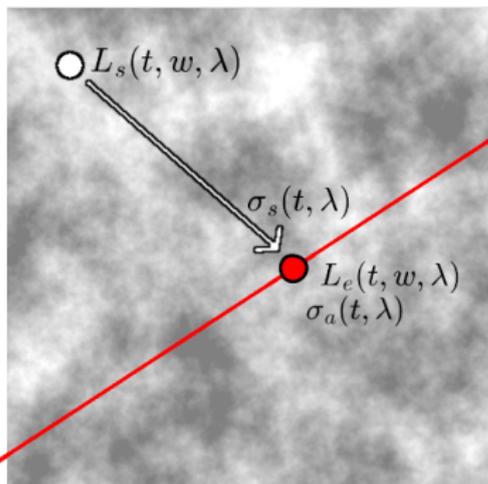
Volumetric rendering equation²



$$L_{local,\lambda}(t, w) = \left[\begin{array}{l} L_s(t, w, \lambda) \sigma_s(t, \lambda) \\ + L_e(t, w, \lambda) \end{array} \right]$$

²Kajiya and Herzen, "Ray Tracing Volume Densities".

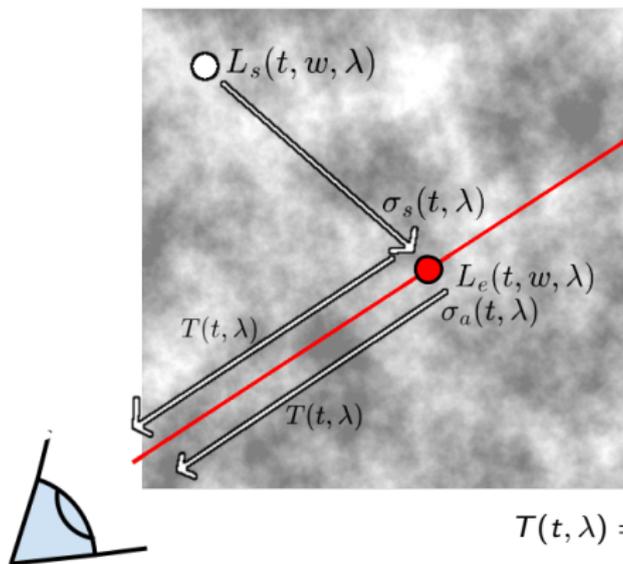
Volumetric rendering equation²



$$L_{local, \lambda}(t, w) = [L_s(t, w, \lambda)\sigma_s(t, \lambda) + L_e(t, w, \lambda)\sigma_a(t, \lambda)]$$

²Kajiya and Herzen, "Ray Tracing Volume Densities".

Volumetric rendering equation²

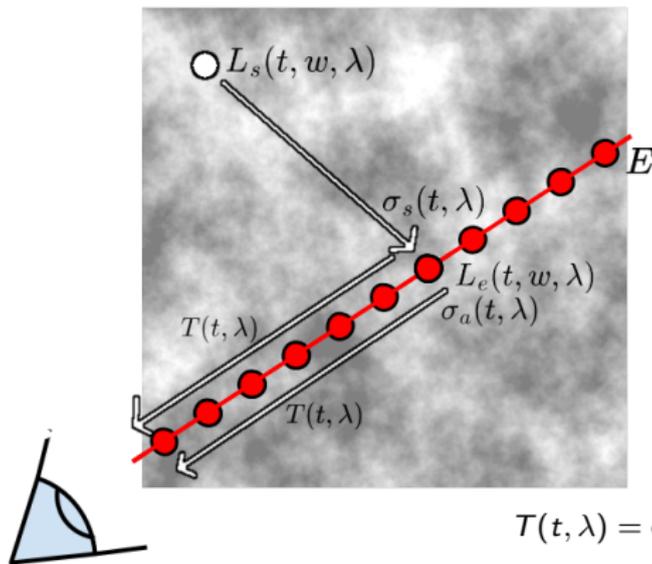


$$T(t, \lambda) = \exp\left(-\int_{u=0}^t \sigma_t(u, \lambda) du\right)$$

$$L_{local, \lambda}(t, w) = T(t, \lambda) [L_s(t, w, \lambda) \sigma_s(t, \lambda) + L_e(t, w, \lambda) \sigma_a(t, \lambda)]$$

²Kajiya and Herzen, "Ray Tracing Volume Densities".

Volumetric rendering equation²



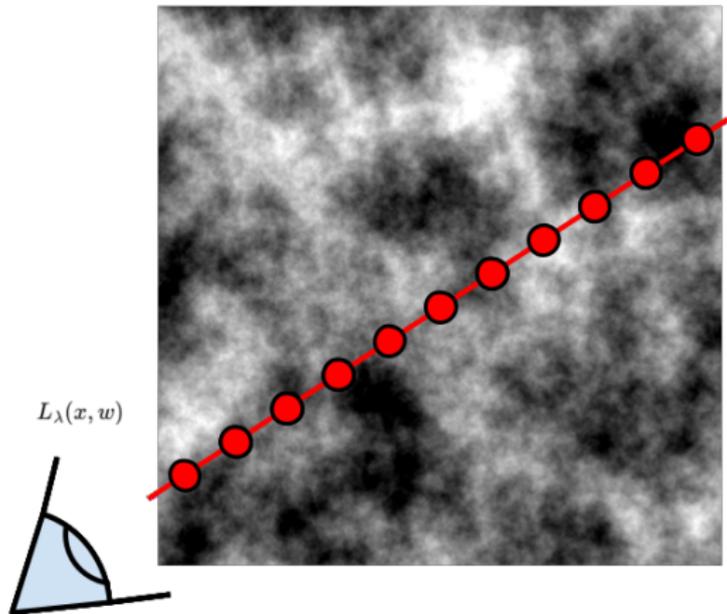
$$T(t, \lambda) = \exp\left(-\int_{u=0}^t \sigma_t(u, \lambda) du\right)$$

$$L_{local, \lambda}(t, w) = T(t, \lambda) [L_s(t, w, \lambda) \sigma_s(t, \lambda) + L_e(t, w, \lambda) \sigma_a(t, \lambda)]$$

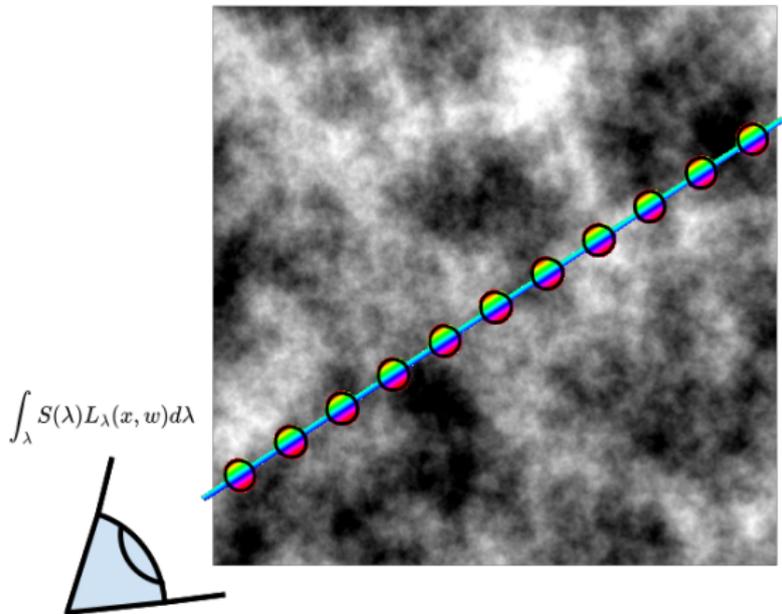
$$L_\lambda(x, w) = \int_t^E L_{local, \lambda}(t, w) dt$$

²Kajiya and Herzen, "Ray Tracing Volume Densities".

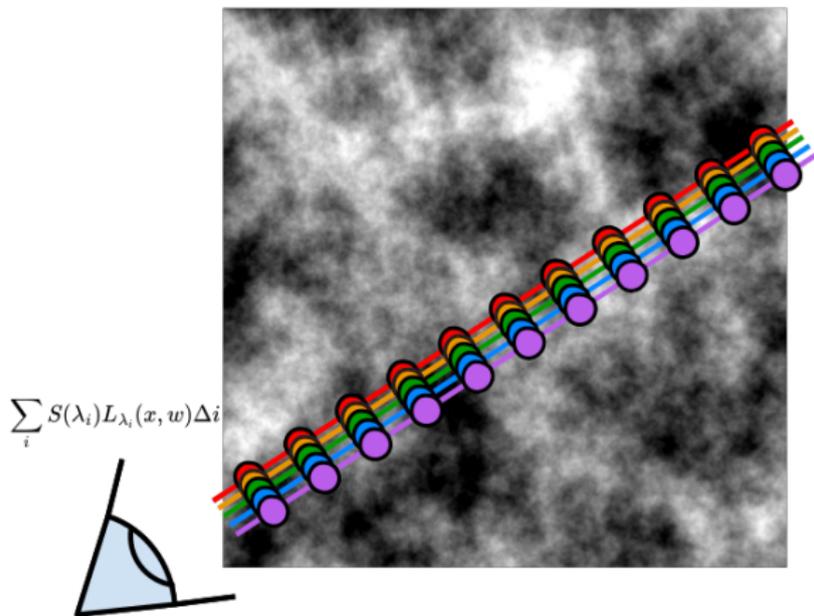
Spectral rendering



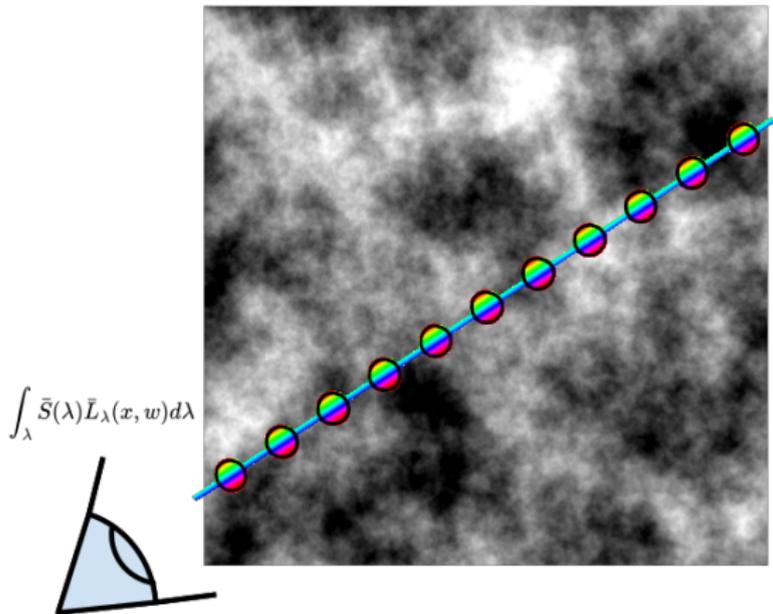
Spectral rendering



Spectral rendering



Spectral rendering



Basis function

$$\int_{\lambda} \int_t f(t, \lambda) dt d\lambda$$

Double numerical integration is generic but costly

³Raso and Fournier, "A Piecewise Polynomial Approach to Shading Using Spectral Distributions"; Peercy, "Linear Color Representations for Full Speed Spectral Rendering"; Peters et al., "Using Moments to Represent Bounded Signals for Spectral Rendering".

Basis function

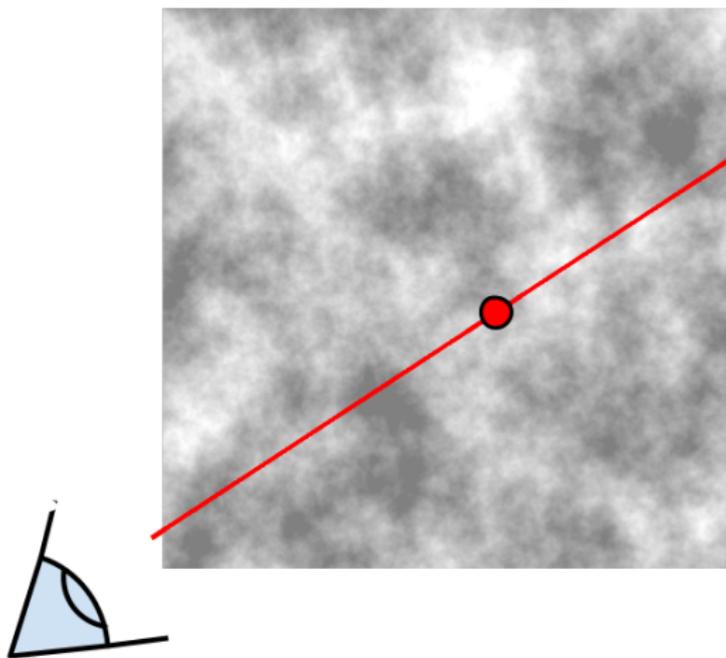
$$\int_{\lambda} \int_t f(t, \lambda) dt d\lambda \approx \int_t \boxed{\sum_i \int_{\lambda} \bar{f}_i(t, \lambda) d\lambda} dt$$

-
- Multiple efficient ways to model spectrum in surface rendering pipelines³ but none for spectral volumetric pipelines (spatial-spectral dual dependency)
 - Requirements: **Stable, Generic, Precise**

³Raso and Fournier, “A Piecewise Polynomial Approach to Shading Using Spectral Distributions”; Peercy, “Linear Color Representations for Full Speed Spectral Rendering”; Peters et al., “Using Moments to Represent Bounded Signals for Spectral Rendering”.

Model

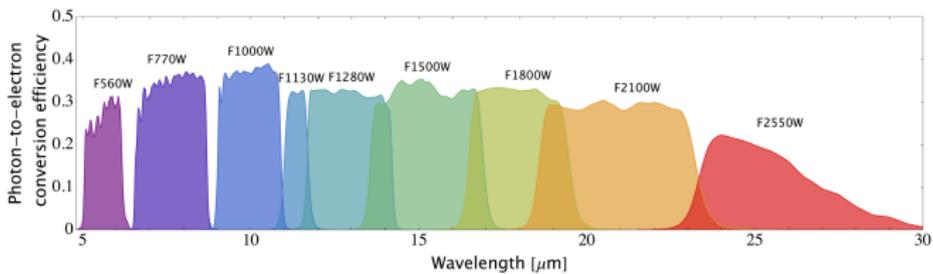
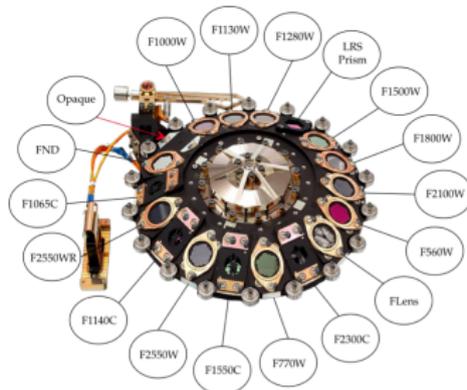
Density field



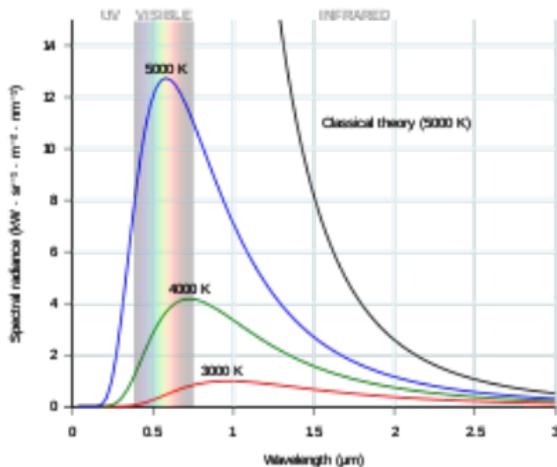
Fractional Brownian
Motion⁴ on Simplex
noise

⁴Decreusefond and Üstünel, "Fractional Brownian motion: theory and applications".

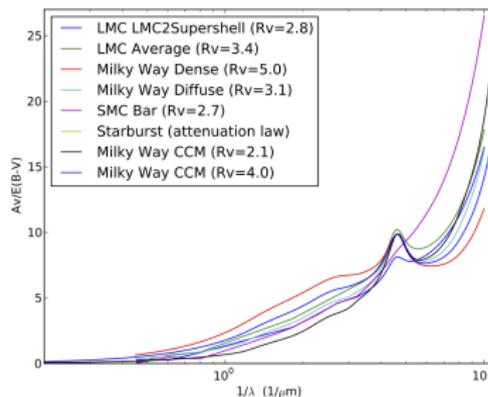
Telescope filters



Other involved spectra



Emission spectrum

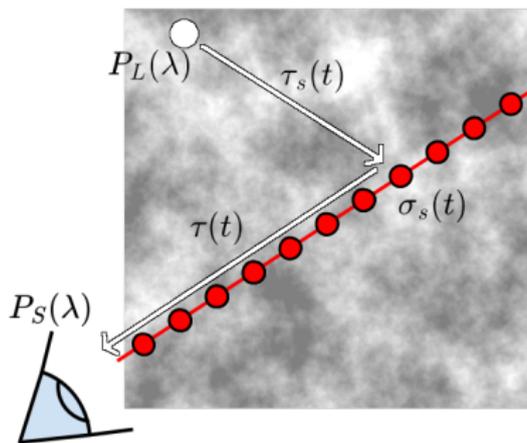


Absorption spectrum

Final model

$$\bar{f}(t, \lambda) = \sigma_i(t)P(\lambda)e^{-g(t)/\lambda}$$

$$g(t) = (\tau(t) + \tau_s(t))C_T$$



$\bar{f}(t, \lambda)$ can be
analytically
integrated in
 $\mathcal{O}(\text{deg}(P))$

Acceleration: exponential projection

Still a lot of computations

$$\int_{\lambda} \int_t f(t, \lambda) dt d\lambda \approx \int_{\lambda} \sum_i f(i, \lambda) \Delta_i d\lambda$$

Still a lot of computations

$$\begin{aligned}\int_{\lambda} \int_t f(t, \lambda) dt d\lambda &\approx \int_{\lambda} \sum_i f(i, \lambda) \Delta_i d\lambda \\ &\approx \int_{\lambda} \sum_i \sigma_s(i) P(\lambda) e^{-g(i)/\lambda} \Delta_i d\lambda\end{aligned}$$

Still a lot of computations

$$\begin{aligned}\int_{\lambda} \int_t f(t, \lambda) dt d\lambda &\approx \int_{\lambda} \sum_i f(i, \lambda) \Delta_i d\lambda \\ &\approx \int_{\lambda} \sum_i \sigma_s(i) P(\lambda) e^{-g(i)/\lambda} \Delta_i d\lambda \\ &= \sum_i \sigma_s(i) \boxed{\int_{\lambda} P(\lambda) e^{-g(i)/\lambda} d\lambda} \Delta_i\end{aligned}$$

i analytical integrals must be computed per ray

Projection on a subset of the space

$$\mathbb{F} = \{ f_a \mid a \in \mathbb{R}^+ \}$$

$$f_a: \mathbb{D} \longrightarrow \mathbb{R}$$

$$\lambda \longmapsto e^{-a/\lambda}$$

Projection on a subset of the space

$$\mathbb{F} = \{ f_a \mid a \in \mathbb{R}^+ \}$$

$$f_a: \mathbb{D} \longrightarrow \mathbb{R}$$

$$\lambda \longmapsto e^{-a/\lambda}$$

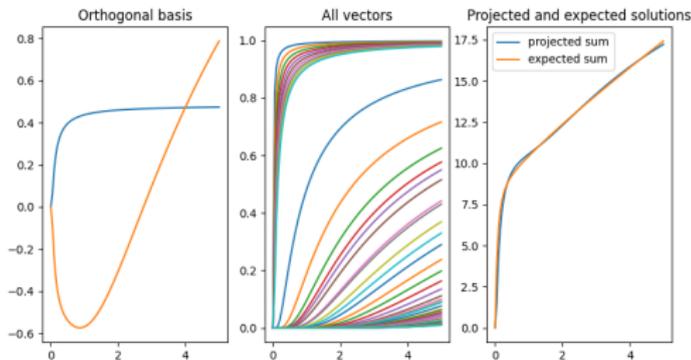
$$\langle f, g \rangle = \int_{\mathbb{D}} f(\lambda)g(\lambda)d\lambda$$

Projection on a subset of the space

$$\mathbb{F} = \{ f_a \mid a \in \mathbb{R}^+ \}$$

$$f_a: \mathbb{D} \longrightarrow \mathbb{R}$$

$$\lambda \longmapsto e^{-a/\lambda}$$



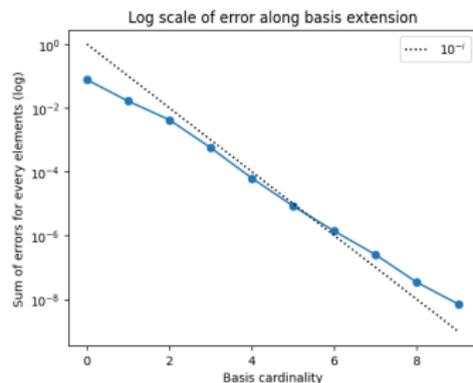
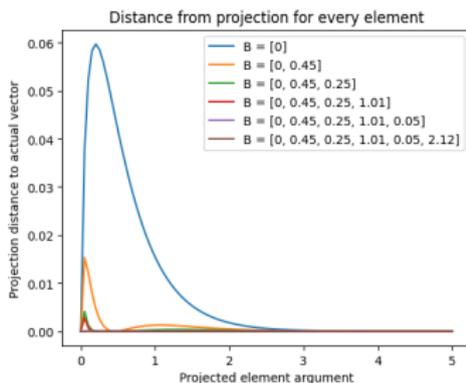
The sum of elements is close to its projection

Results

Best basis B is obtained by **minimizing** $\sum_{f \in \mathbb{F}} \|f - \text{proj}_B(f)\|$

Results

Best basis B is obtained by **minimizing** $\sum_{f \in \mathbb{F}} \|f - \text{proj}_B(f)\|$



$$D = [0, 1\mu\text{m}]$$

Adding an element to the basis divides the error by 10.

Projection on a subset of the space

Projecting the elements on a subset of the space will **reduce the number of integrals to compute**

$$e^{-g(i)/\lambda} \approx A_i e^{-d_a/\lambda} + B_i e^{-d_b/\lambda}$$
$$\sum_i e^{-g(i)/\lambda} \approx \left(\sum_i A_i\right) e^{-d_a/\lambda} + \left(\sum_i B_i\right) e^{-d_b/\lambda}$$

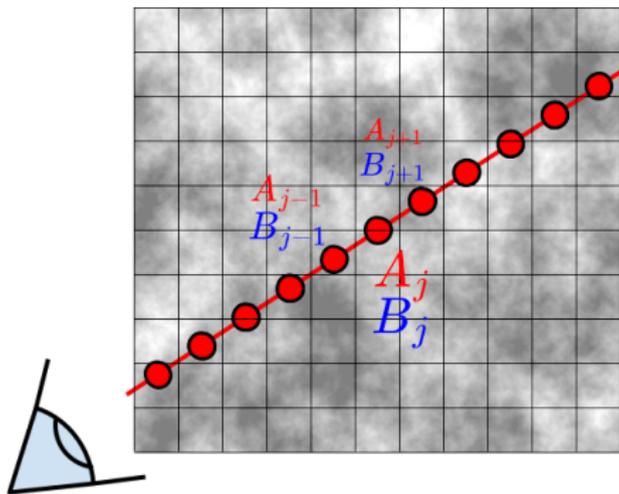
Projection on a subset of the space

Projecting the elements on a subset of the space will **reduce the number of integrals to compute**

$$e^{-g(i)/\lambda} \approx A_i e^{-d_a/\lambda} + B_i e^{-d_b/\lambda}$$
$$\sum_i e^{-g(i)/\lambda} \approx \left(\sum_i A_i\right) e^{-d_a/\lambda} + \left(\sum_i B_i\right) e^{-d_b/\lambda}$$

-
- $e^{-d_a/\lambda}$ and $e^{-d_b/\lambda}$ chosen as the most informative elements of the family.
 - Basis constructed with Gram-Schmidt

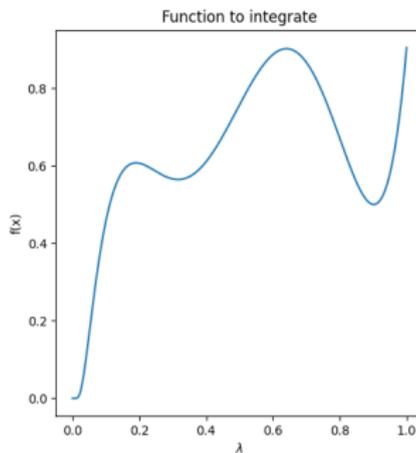
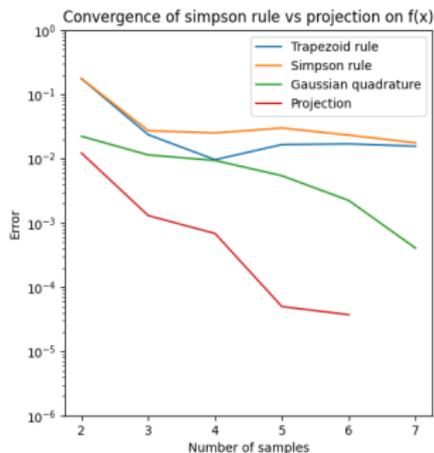
Algorithm in practice



$$\sum_j P(\lambda)h(t_j, w)(A_j e^{-d_a/\lambda} + B_j e^{-d_b/\lambda})$$

Convergence of methods

$$f(x) = ((x - 1)(x - .2)(x - .5)(x - .7)(x)100 + 1)e^{-.1/x}$$



The projection method converges faster than Newtons or Gauss numerical methods.

Numerical complexity

Double Netwon integration: $\mathcal{O}(N_{space} \cdot (M_{wavelength} + N_{space}))$

Naive analytical spectral integration: $\mathcal{O}(N_{space} \cdot (Deg(P) + N_{space}))$

Run-time integration: $\mathcal{O}(N_{space} \cdot Card_{basis} + Deg(P) \cdot Card_{basis})$

⁵ $N=100, M=30, C_b=3, Deg(P)=5$, GPU= Nvidia GTX 1080 Ti, scene A

Numerical complexity

Double Newton integration: $\mathcal{O}(N_{space} \cdot (M_{wavelength} + N_{space}))$

Naive analytical spectral integration: $\mathcal{O}(N_{space} \cdot (Deg(P) + N_{space}))$

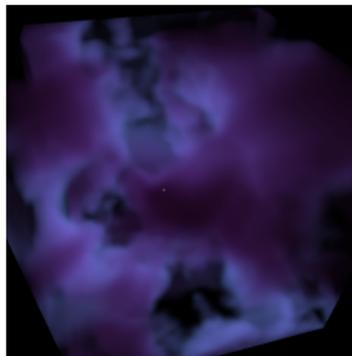
Run-time integration: $\mathcal{O}(N_{space} \cdot Card_{basis} + Deg(P) \cdot Card_{basis})$

Method	Avg Time per frame ⁵
Newton	260ms
Naive analytical computation	160ms
Projection	30ms

All spectra are free variables, as well as the camera position and angle.

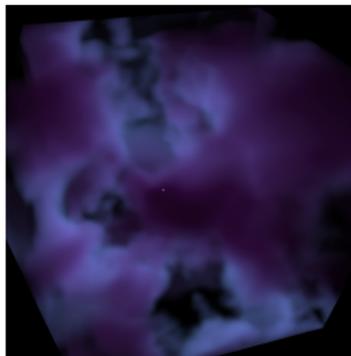
⁵ $N=100, M=30, C_b=3, Deg(P)=5$, GPU= Nvidia GTX 1080 Ti, scene A

Results



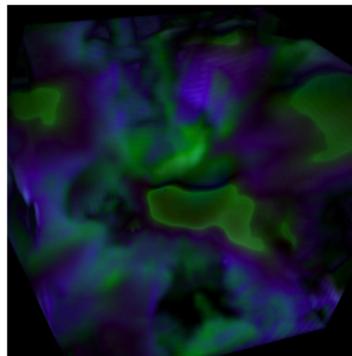
Double Network
Integration
 $M = 30, N = 100$

260 ms



Approximation by
projection
 $D(P) = 5, C_B = 3$

30 ms



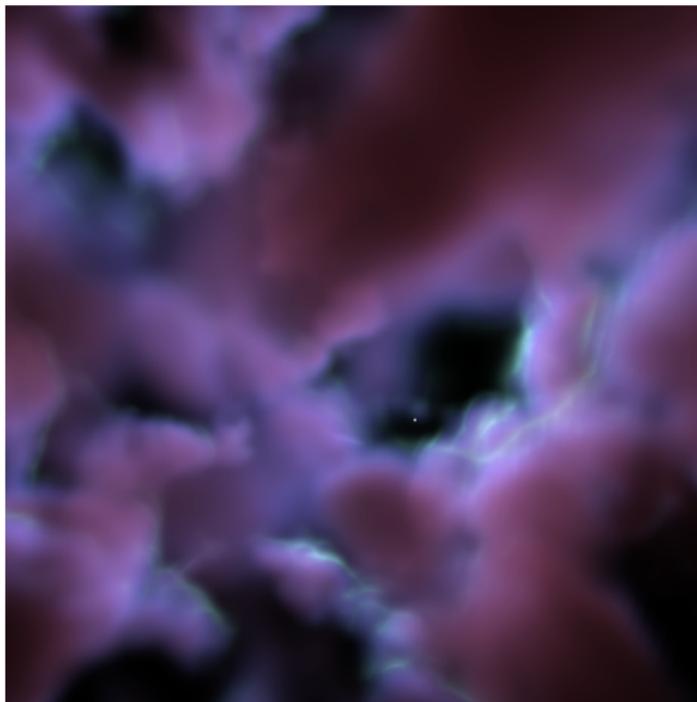
Absolute difference
 $\times 16$

Summary:

- Fast, precise integration
- Explorable model
- Scalable in terms of error
- Extendable for multiple lights

Future works:

- Better spatial integral
- Relaxation of some of our constraints (phase, attenuation profile...)
- Multiple bounce integration



Thanks for listening!

References



Decreusefond, Laurent and Ali Üstünel. "Fractional Brownian motion: theory and applications". In: <http://dx.doi.org/10.1051/proc:1998014> 5 (Jan. 1999). DOI: 10.1051/proc:1998014.



Engel, Klaus, Martin Kraus, and Thomas Ertl. "High-Quality Pre-Integrated Volume Rendering Using Hardware-Accelerated Pixel Shading". In: *Proceedings of the ACM SIGGRAPH*. New York, NY, USA: Association for Computing Machinery, 2001. DOI: 10.1145/383507.383515.



INRIA ANR Galaxy/veRTIGE.

<https://www.inria.fr/fr/une-promenade-virtuelle-dans-notre-galaxie>. Accessed: 2023-06-14.



Kajiya, James and Brian von Herzen. "Ray Tracing Volume Densities". In: *ACM SIGGRAPH Computer Graphics* 18 (July 1984), pp. 165–174. DOI: 10.1145/964965.808594.



Peercy, Mark S. "Linear Color Representations for Full Speed Spectral Rendering". In: *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '93. Anaheim, CA: Association for Computing Machinery, 1993, 191–198. DOI: 10.1145/166117.166142.



Peters, Christoph et al. "Using Moments to Represent Bounded Signals for Spectral Rendering". In: *ACM Trans. Graph.* 38.4 (2019). DOI: 10.1145/3306346.3322964.



Raso, M. G. and A. Fournier. "A Piecewise Polynomial Approach to Shading Using Spectral Distributions". In: *Proceedings of Graphics Interface '91*. GI '91. Calgary, Alberta, Canada: Canadian Man-Computer Communications Society, 1991, pp. 40–46.

Appendix

TOC

- 6 bad approx: SVRE pre-integrated
- 7 Multiple spectra
- 8 Multiple spectra in scene
- 9 Complexity
- 10 Comparison between pictures
- 11 Argument range

Spectral pre-integration

- Traditional pre-integration⁶ cannot be directly applied to our problem, due to the spectrum which isn't stable under product.
- Approximation: $\int_{\lambda=\lambda_1}^{\lambda_2} f(t, \lambda) d\lambda \approx \sum_k C_k(\lambda_1, \lambda_2) e^{d(t, \lambda_1, \lambda_2)}$

⁶Engel, Kraus, and Ertl, "High-Quality Pre-Integrated Volume Rendering Using Hardware-Accelerated Pixel Shading".

Spectral pre-integration

- Traditional pre-integration⁶ cannot be directly applied to our problem, due to the spectrum which isn't stable under product.
- Approximation: $\int_{\lambda=\lambda_1}^{\lambda_2} f(t, \lambda) d\lambda \approx \sum_k C_k(\lambda_1, \lambda_2) e^{d(t, \lambda_1, \lambda_2)}$

$$\sum_{k=0}^N P_k \int_{\lambda=\lambda_1}^{\lambda_2} f_k(d(t), \lambda) d\lambda$$

$$\approx \left[\sum_{k=0}^N P_k C_k(\lambda_1, \lambda_2) \exp(-[\sigma_t(0)\Delta t] \cdot \epsilon(\lambda_1, \lambda_2)) \right]_1 \cdot \left[\prod_{i=1}^t \exp(-\sigma_t(i)\Delta_j \cdot \epsilon(\lambda_1, \lambda_2)) \right]_2 \left[\exp(-\tau_s(t) \cdot \epsilon(\lambda_1, \lambda_2)) \right]_3 \quad (1)$$

⁶Engel, Kraus, and Ertl, "High-Quality Pre-Integrated Volume Rendering Using Hardware-Accelerated Pixel Shading".

Multiple spectra

Pre-process, projection of light voxel:

$$\sum_i P_i(\lambda) e^{-\tau_s(t)/\lambda} = aP_A(\lambda) e^{-\tau_1/\lambda} + bP_B(\lambda) e^{-\tau_2/\lambda}$$

Run-time projection of current optical depth:

$$e^{-\tau/\lambda} = ce^{-\tau_1/\lambda} + de^{-\tau_2/\lambda}$$

Then:

$$\sum_i P_i(\lambda) e^{-(\tau(t) + \tau_s(t))/\lambda} \tag{2}$$

$$= acP_A(\lambda) e^{-2\tau_1/\lambda} + adP_A(\lambda) e^{-(\tau_2 + \tau_1)/\lambda} \tag{3}$$

$$+ bcP_B(\lambda) e^{-(\tau_1 + \tau_2)/\lambda} + bdP_B(\lambda) e^{-2\tau_2/\lambda} \tag{4}$$

Multiple spectra in scene

Spectral-spatial basis propagation

$$\sum_j \sum_i \sigma_s(i) e^{-d(i)/\lambda} P_j(\lambda) \approx A e^{-d_a/\lambda} P_a(\lambda) + B e^{-d_b/\lambda} P_b(\lambda)$$

Spectral-spatial basis propagation

$$\sum_j \sum_i \sigma_s(i) e^{-d(i)/\lambda} P_j(\lambda) \approx A e^{-d_a/\lambda} P_a(\lambda) + B e^{-d_b/\lambda} P_b(\lambda)$$

Double Network integration:

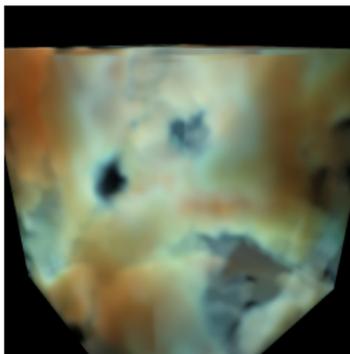
$\mathcal{O}(N_{space} \cdot (M_{wavelength} + N_{space}) \cdot L_{lights})$

Naive analytical spectral integration: $\mathcal{O}(N_{space} \cdot (Deg(P) + N_{space}))$

Projection: $\mathcal{O}(N_{space} \cdot Deg(P) + Deg(P) \cdot Card_{basis})$

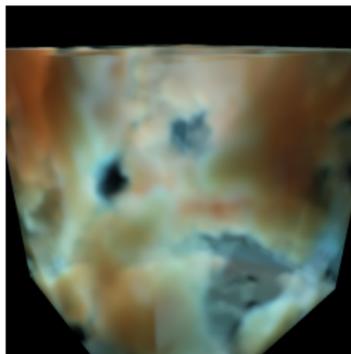
- Loss of source-specific information, phase function must be approximated as isotropic

Results



Double Network
Integration
 $M = 30, N = 100$

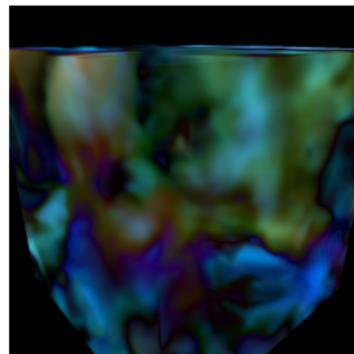
2500 ms



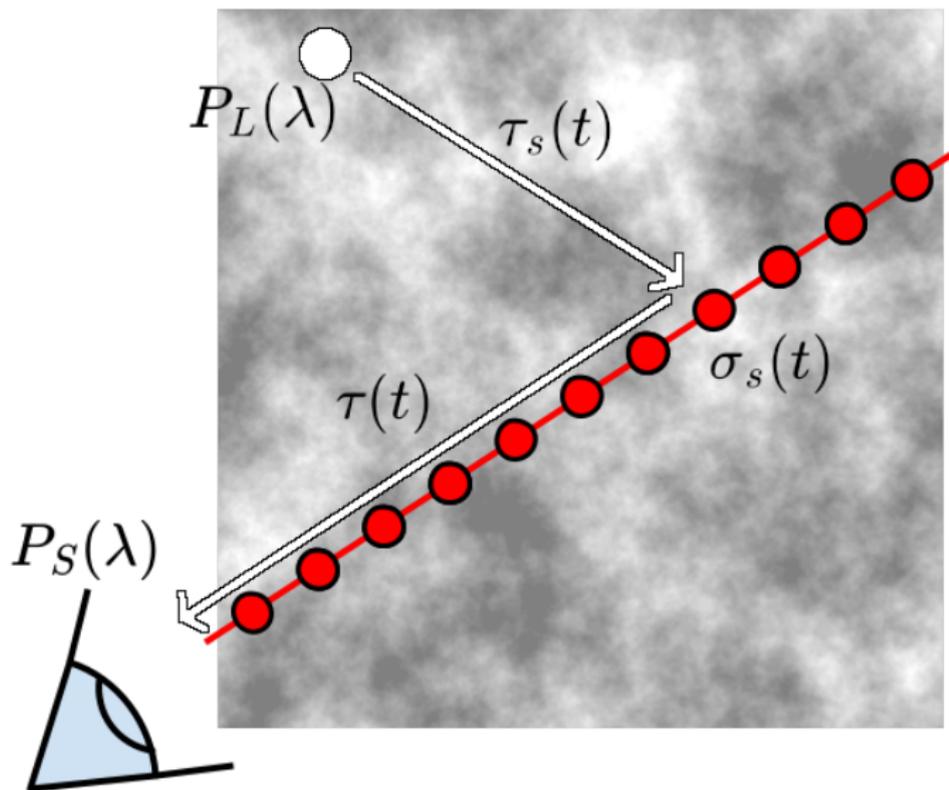
Approximation by
projection

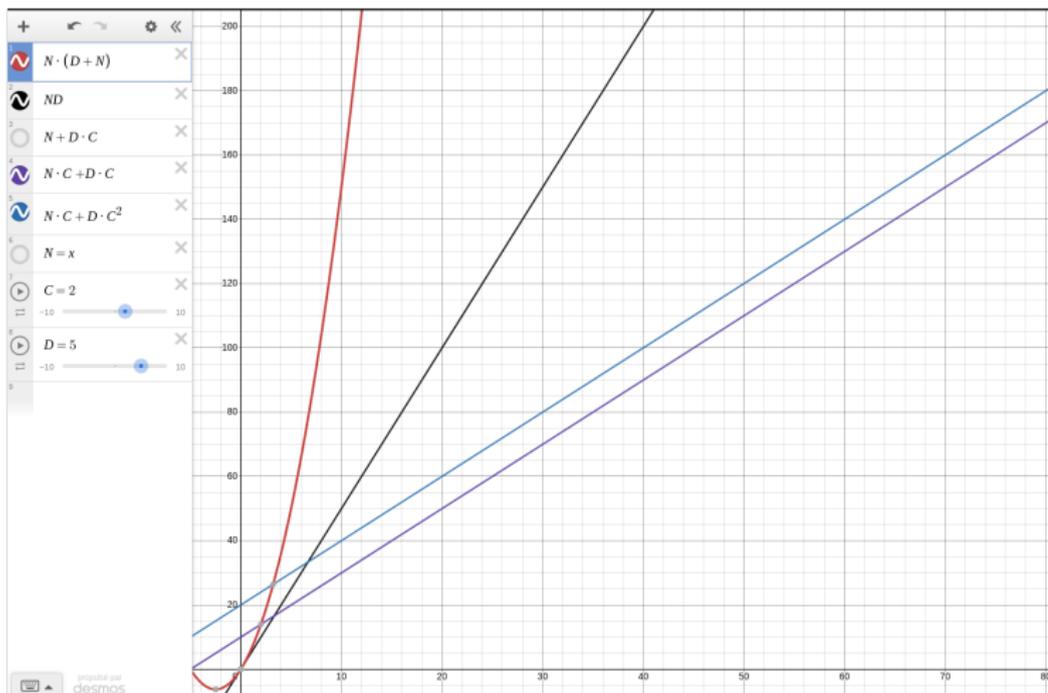
$$L_{lights} = 6, D(P) = 5, C_B = 4$$

30 ms

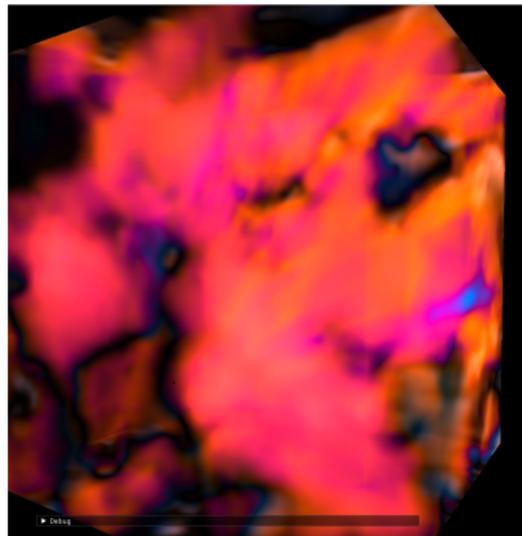
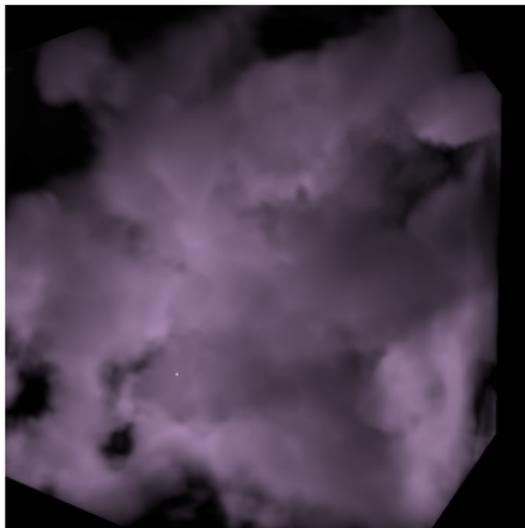


Error x 6

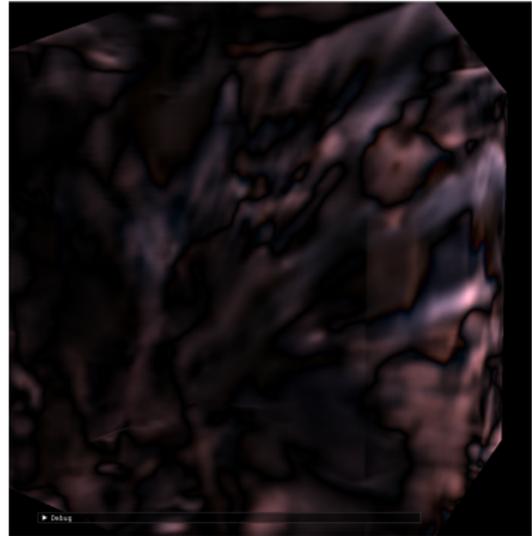
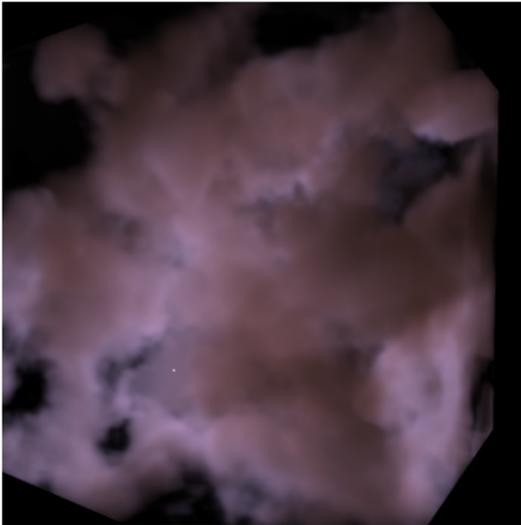




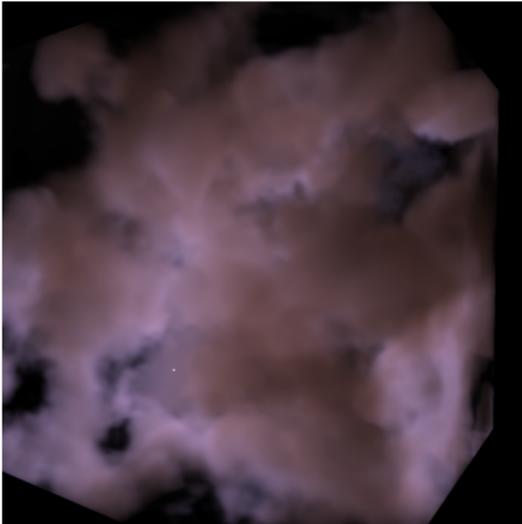
Basis augmentation



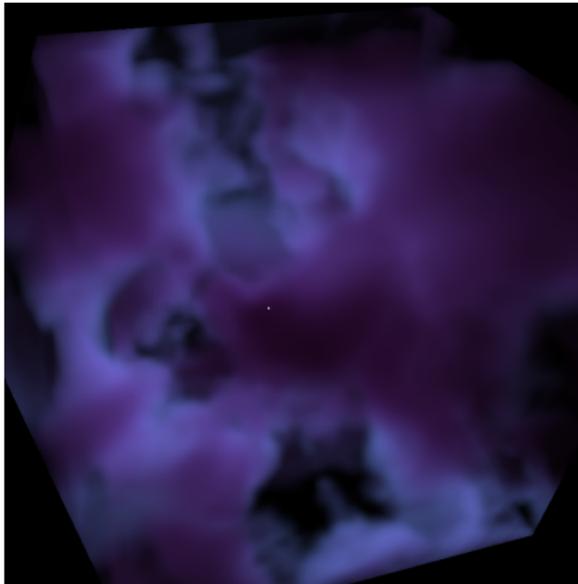
Basis augmentation



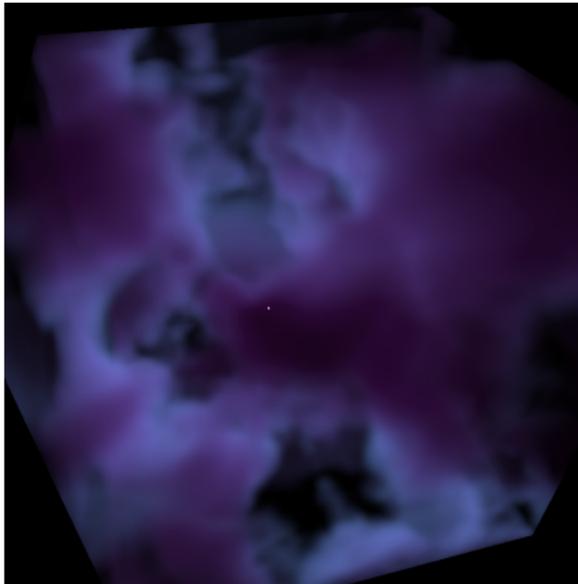
Basis augmentation



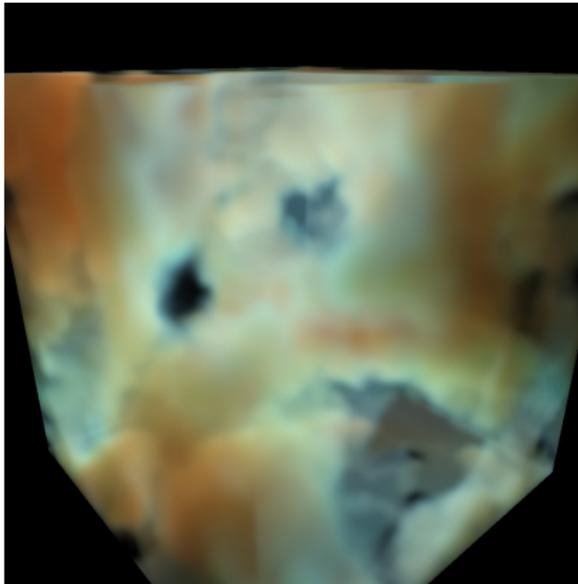
Density results



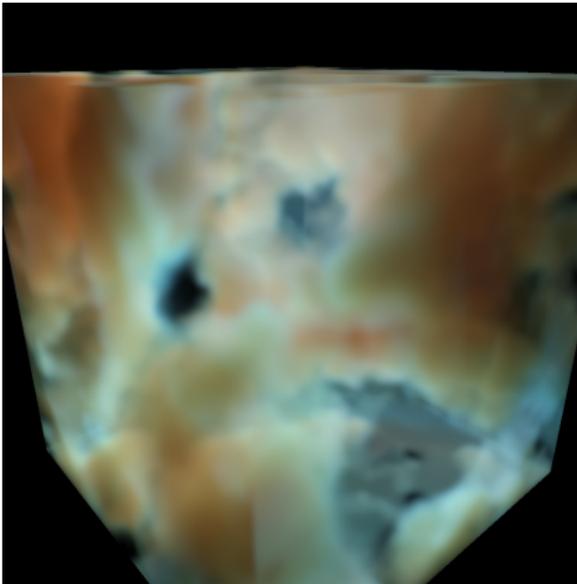
Density results



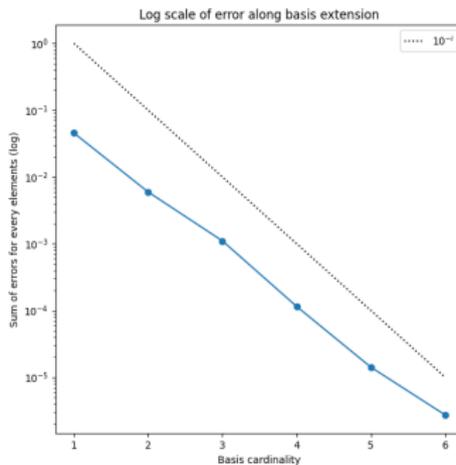
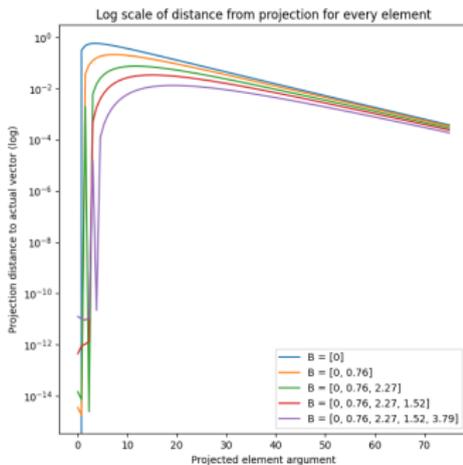
Multiple lights results



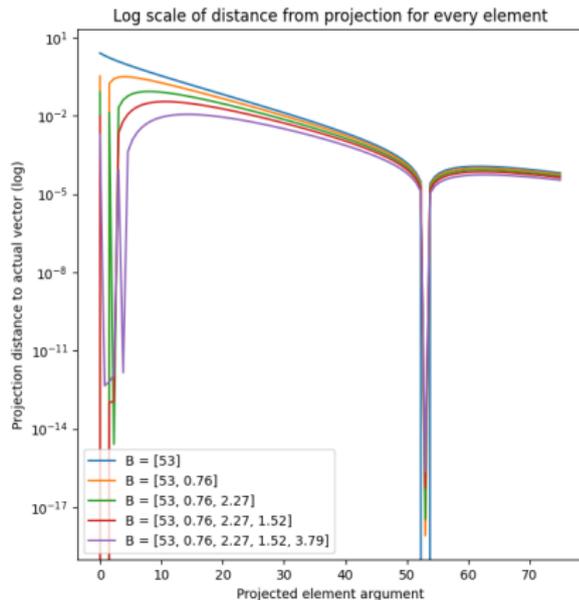
Multiple lights results



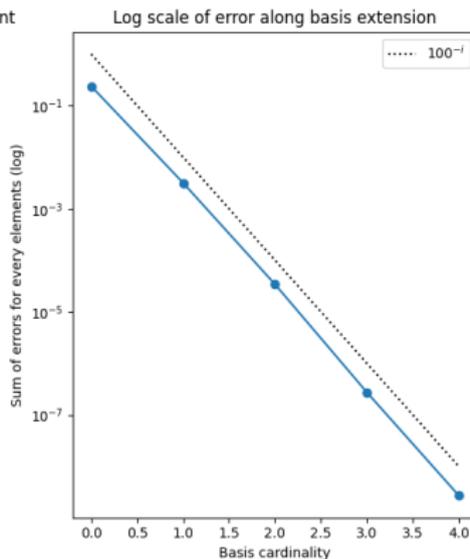
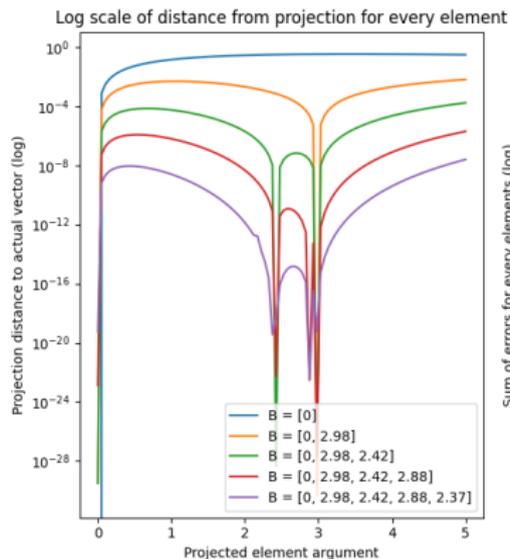
Extended argument range



Trying a huge argument



Increased precision over higher wavelengths



$D = [1\mu\text{m}, 10\mu\text{m}]$