# Paper presentation

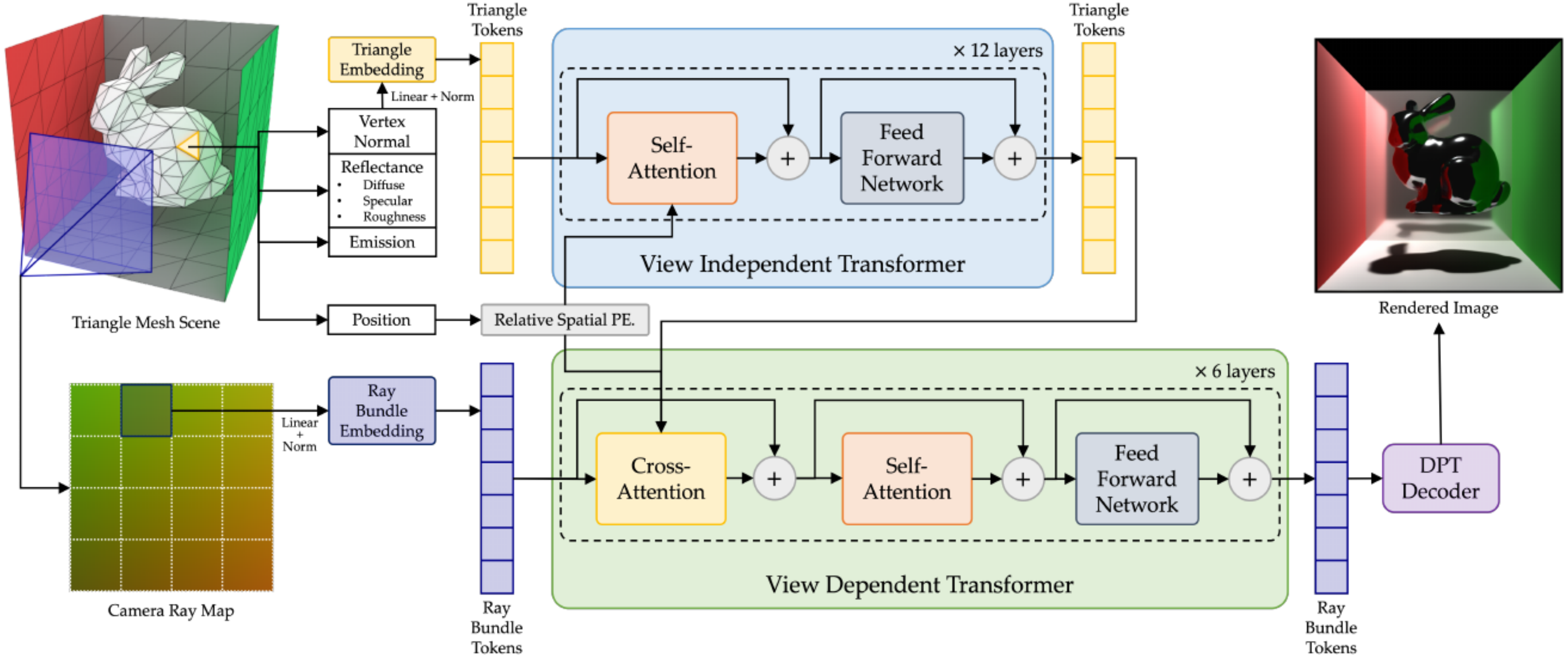
Real-Time Underwater Spectral Rendering

February 2024, Nestor Monzon

Group 1
Course: CS580
Speakers: Daniel Mocanu, Valentin Le Lièvre

# Review of previous lecture

## RenderFormer





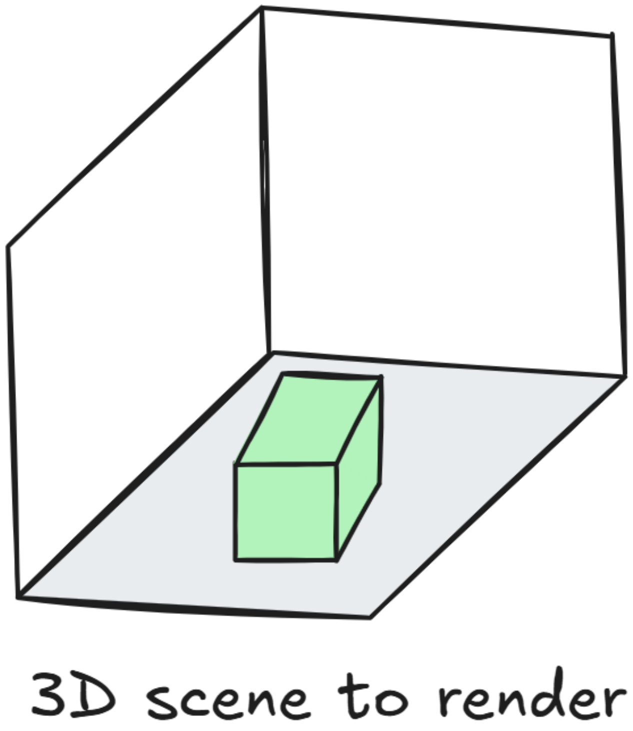

Sources : Microsoft - Render former ; https://microsoft.github.io/renderformer/

# Overview

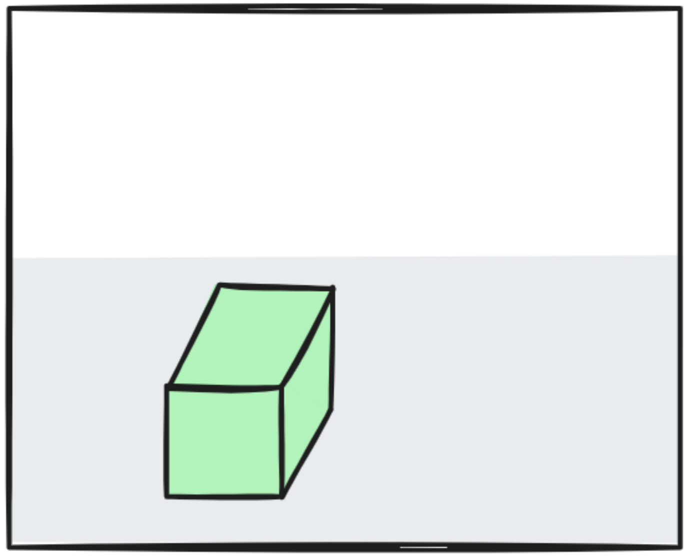
1. Intuitive explanation of spectral rendering
2. Limitation of current method
3. Paper approach (Single Scattering and Multi-Scattering)
4. Jerlov water types and downwelling attenuation
5. Paper solutions to simplify the rendering equations
6. Results

# Intuitive explanation of spectral rendering

# Intuitive explanation of spectral rendering

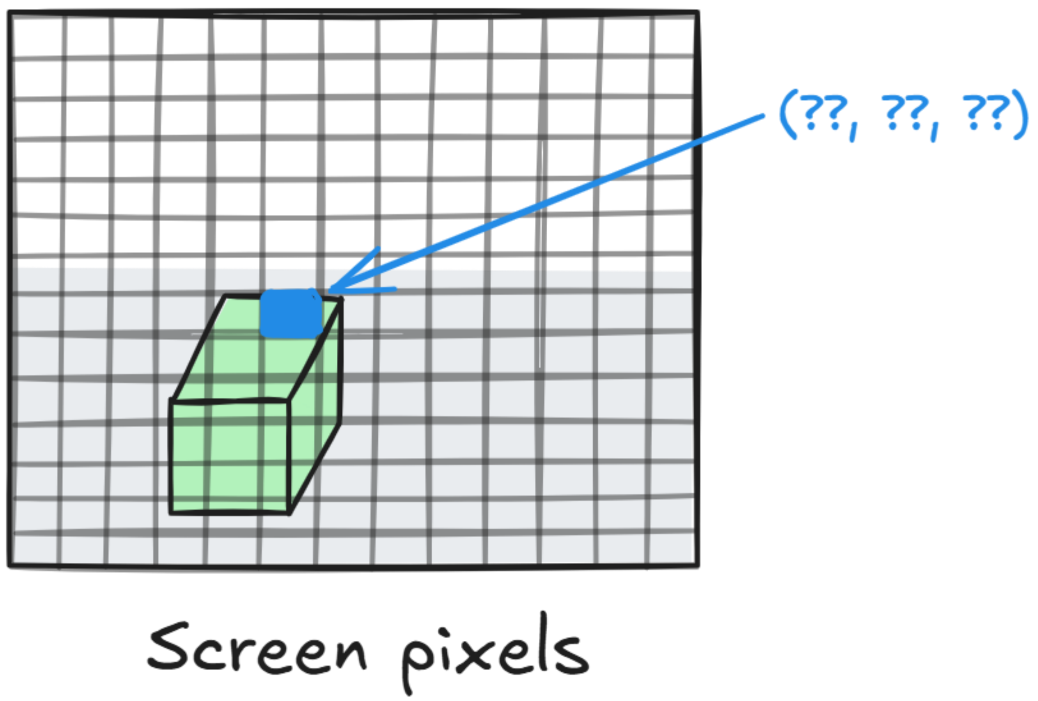


# Intuitive explanation of spectral rendering

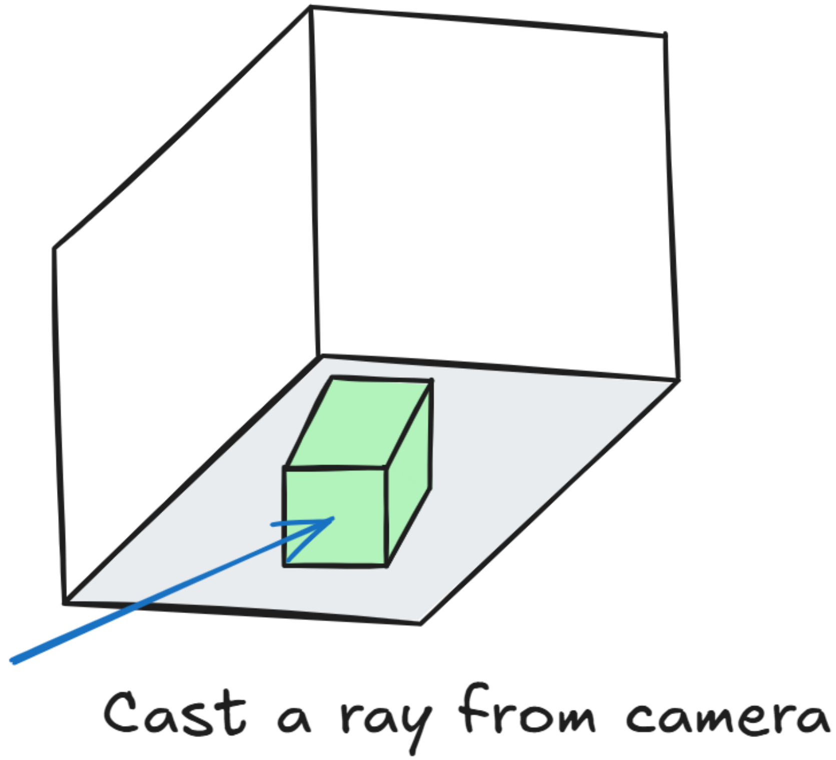


From camera perspective

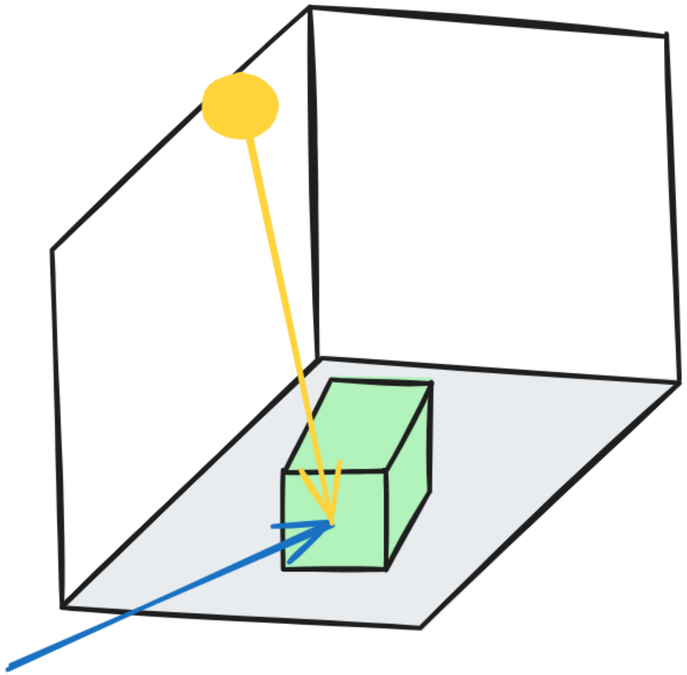
# Intuitive explanation of spectral rendering



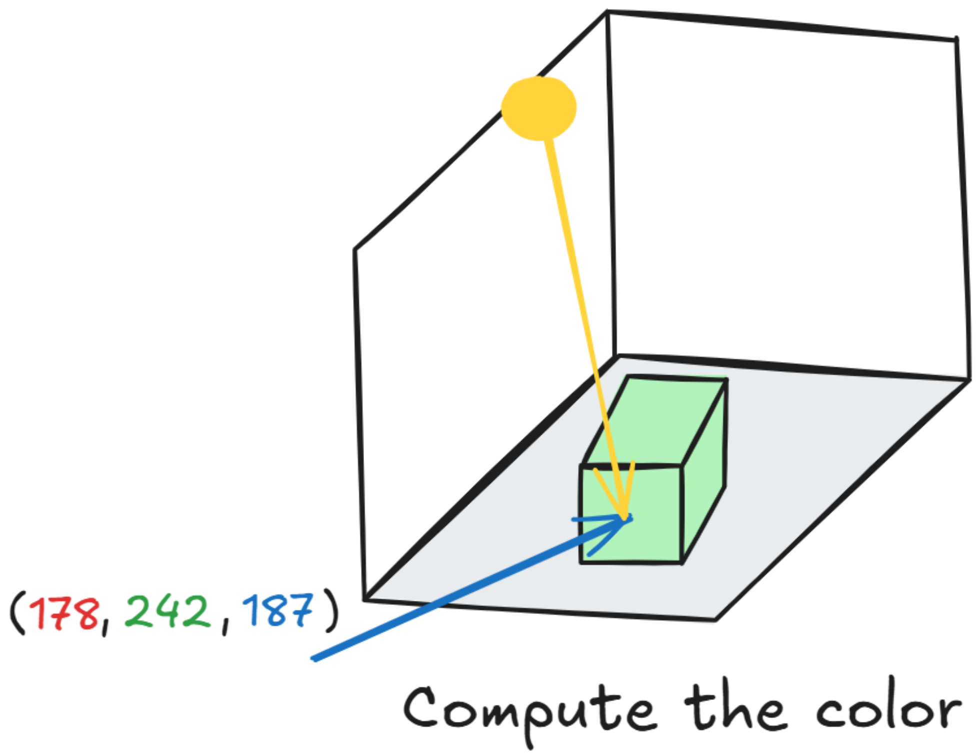
# Intuitive explanation of spectral rendering



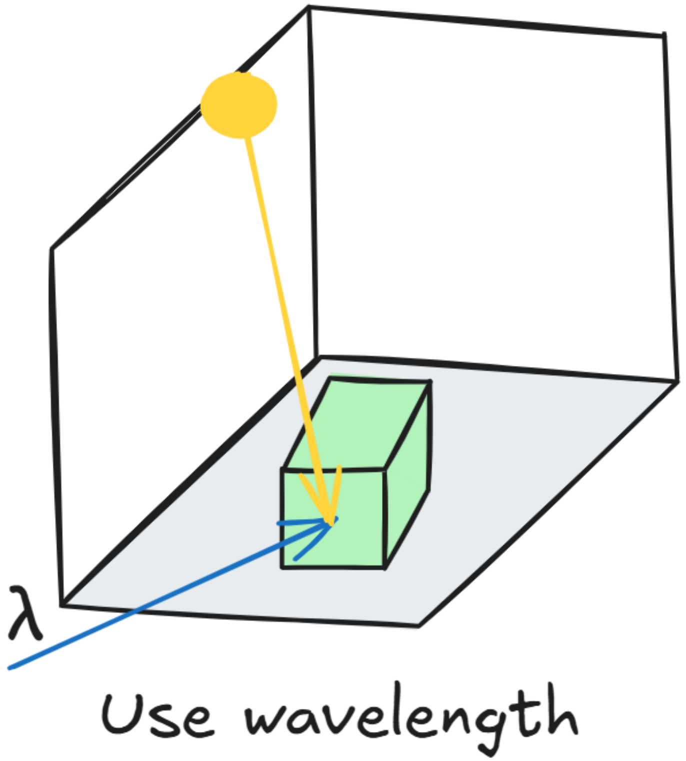
# Intuitive explanation of spectral rendering



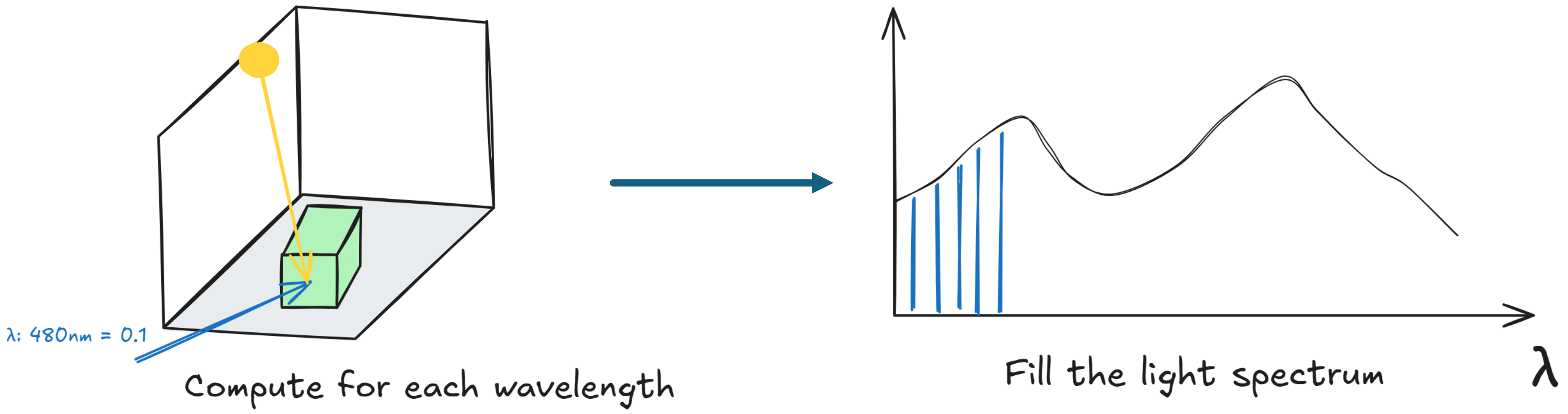
# Intuitive explanation of spectral rendering



# Intuitive explanation of spectral rendering



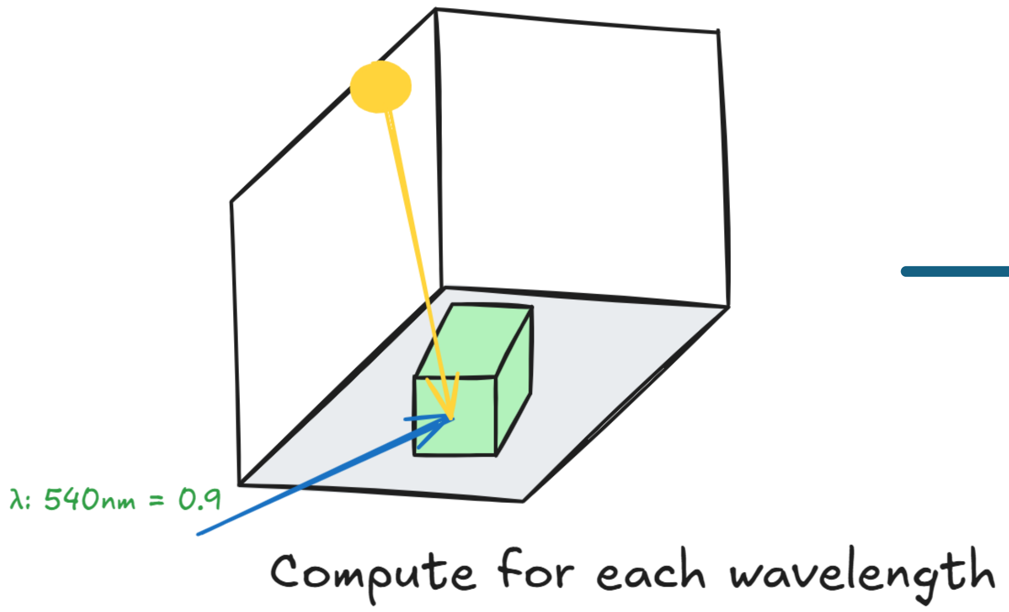
# Intuitive explanation of spectral rendering

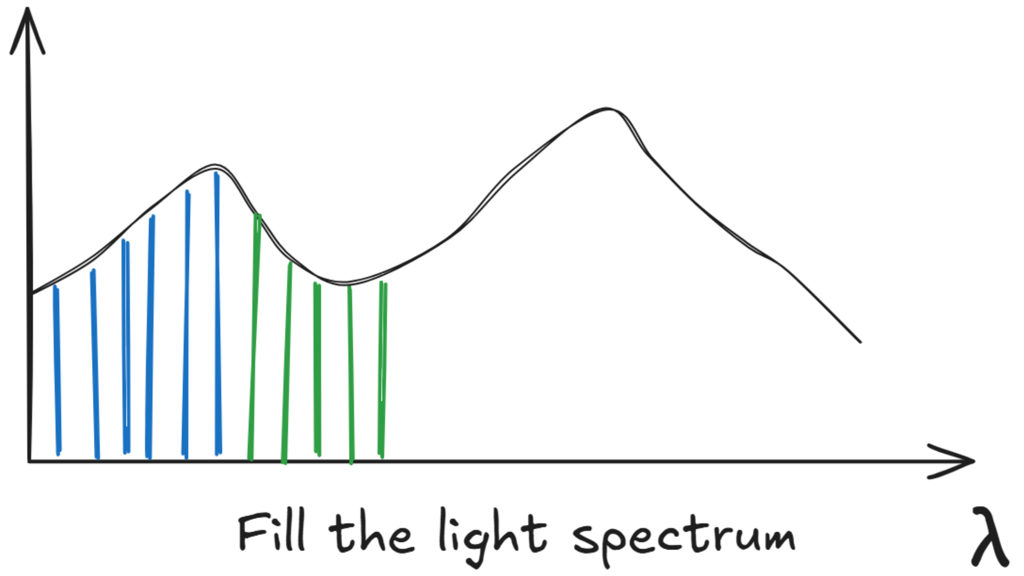



Rendering equation for each wavelength :

$$L_o(x, \omega_o, \lambda) = L_e(x, \omega_o, \lambda) + \int_\Omega f_r(x, \omega_i, \omega_o, \lambda)\ L_i(x, \omega_i, \lambda)\ \cos\theta_i\ d\omega_i$$

# Intuitive explanation of spectral rendering

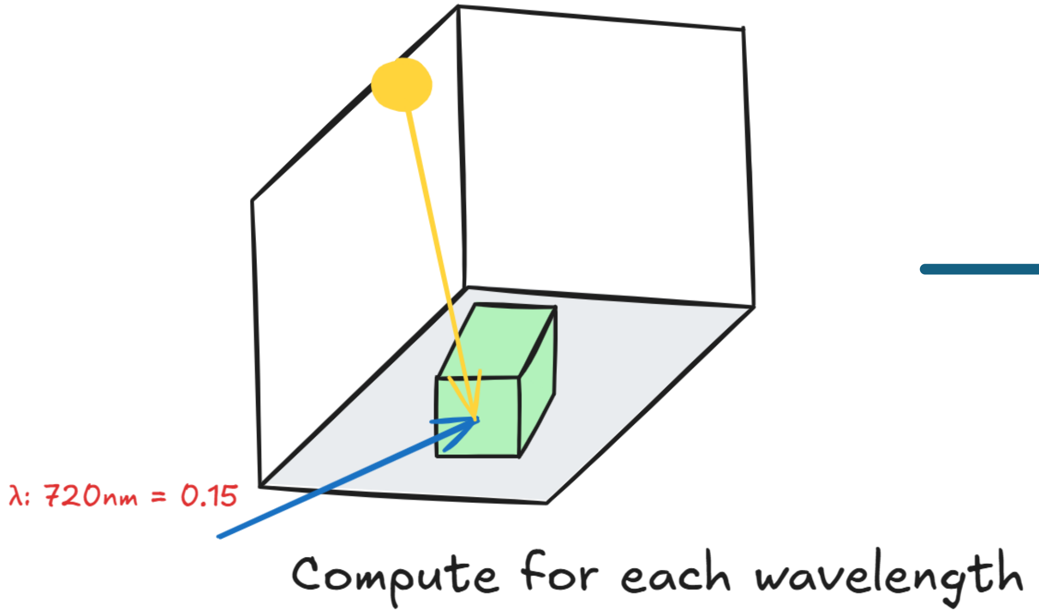


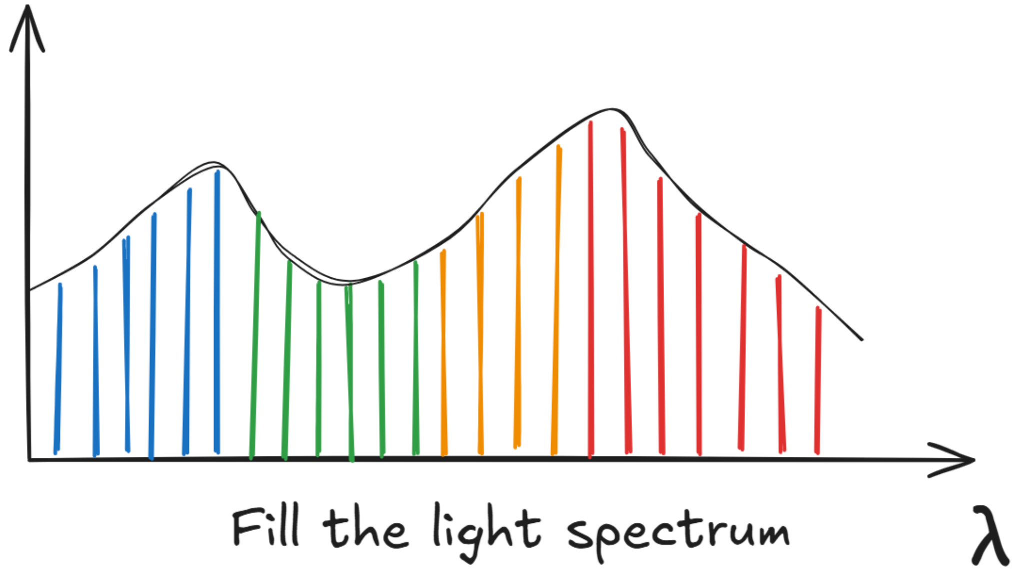




Rendering equation for each wavelength :

$$L_o(x, \omega_o, \lambda) = L_e(x, \omega_o, \lambda) + \int_{\Omega} f_r(x, \omega_i, \omega_o, \lambda)\ L_i(x, \omega_i, \lambda)\ \cos\theta_i\ d\omega_i$$

# Intuitive explanation of spectral rendering

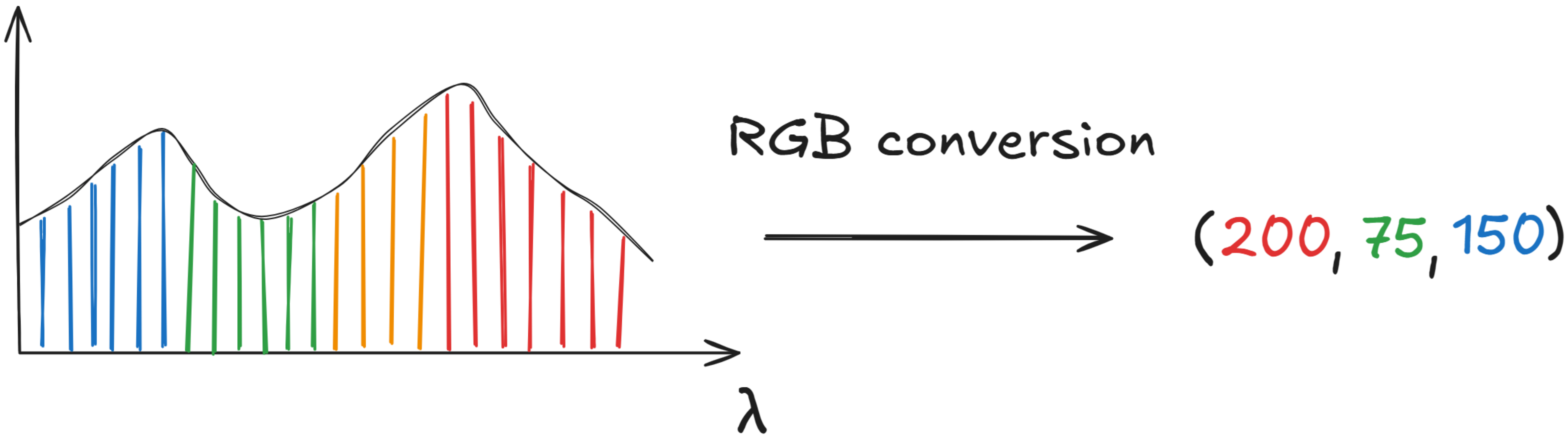


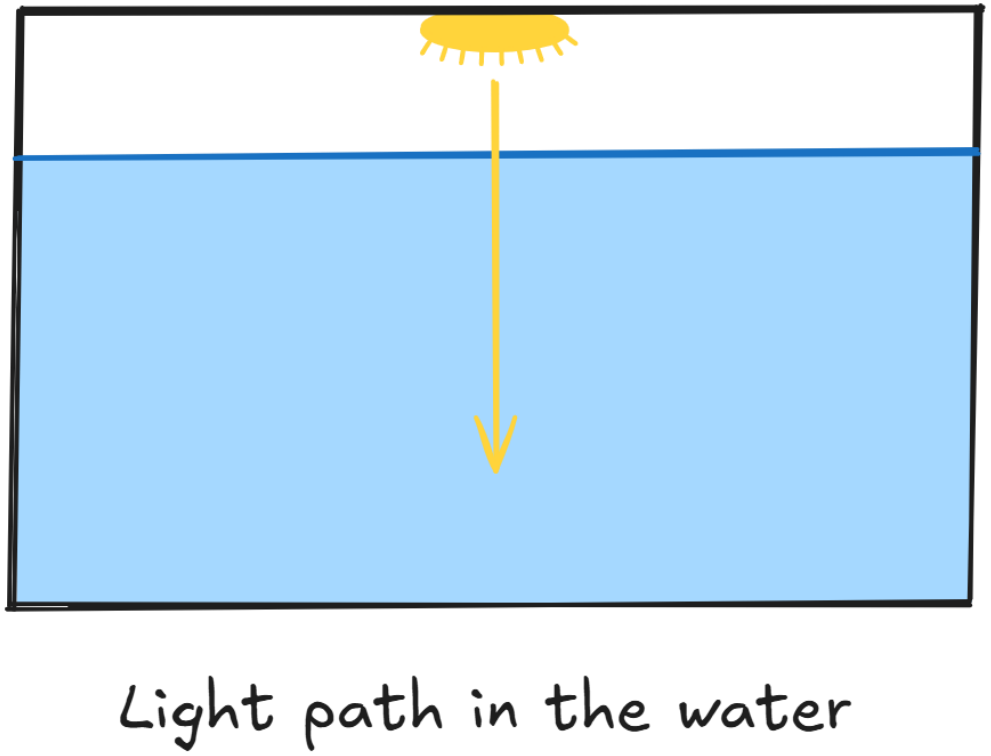

Rendering equation for each wavelength :

$$L_o(x, \omega_o, \lambda) = L_e(x, \omega_o, \lambda) + \int_\Omega f_r(x, \omega_i, \omega_o, \lambda)\ L_i(x, \omega_i, \lambda)\ \cos\theta_i\ d\omega_i$$

# Intuitive explanation of spectral rendering



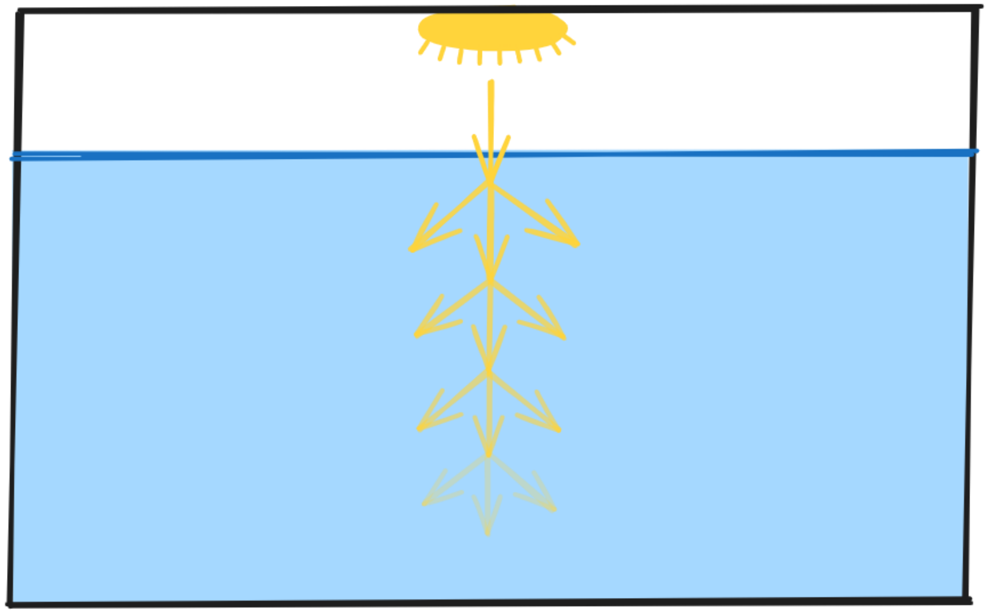
# Intuitive explanation of spectral rendering



# Intuitive explanation of spectral rendering

Scattering Coefficient $\boldsymbol{\delta_s}$ - Probability of light being scattered per meter.

$\boldsymbol{\delta_s L(x_z, \omega)}$ - Differential scattered radiance.

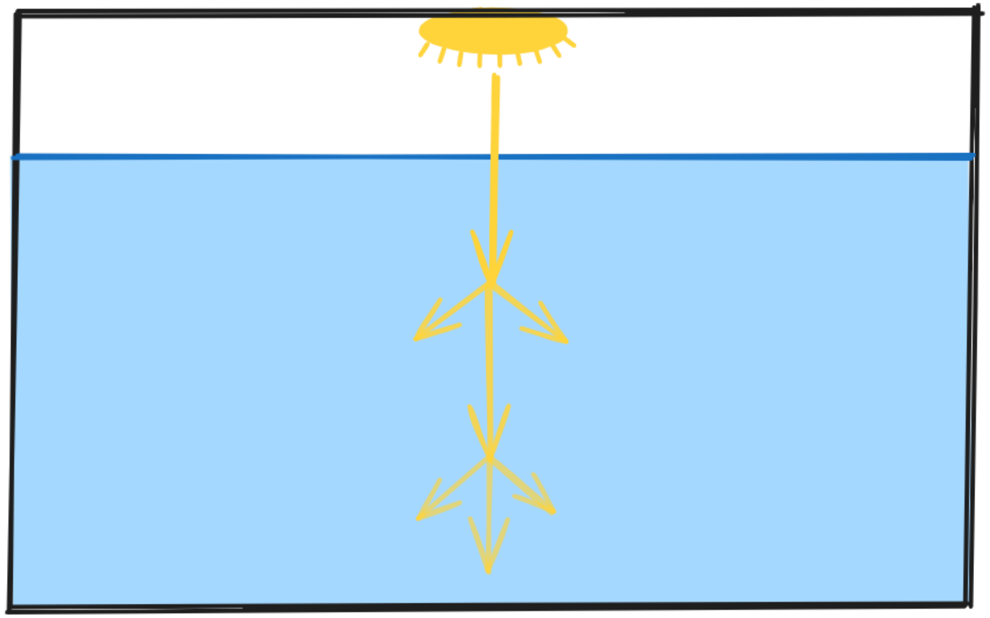


Light scattering

# Intuitive explanation of spectral rendering

Scattering Coefficient $\boldsymbol{\delta_s}$ - Probability of light being scattered per meter.

$\boldsymbol{\delta_s L(x_z, \omega)}$ - Differential scattered radiance.

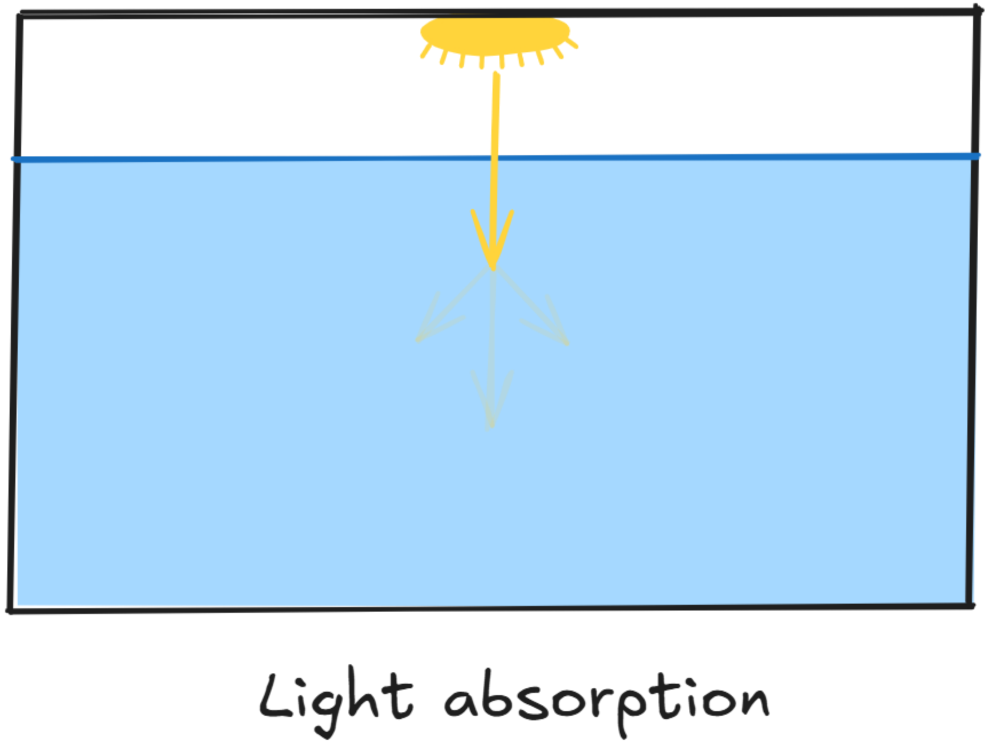


Light scattering

# Intuitive explanation of spectral rendering

Absorption Coefficient $\boldsymbol{\delta_a}$ - Probability of light being absorbed per meter.

$\boldsymbol{\delta_a L(x_z, \omega)}$ - Differential absorbed radiance.



# Intuitive explanation of spectral rendering

**Extinction Coefficient :** $\boldsymbol{\delta}_{\mathrm{t}} = \boldsymbol{\delta}_{\mathrm{a}} + \boldsymbol{\delta}_{s}$ - Probability of absorption or scattering

**Differential Total Extinction :** $\frac{\boldsymbol{dL_{extinction}(x,\omega)}}{\boldsymbol{dz}} = -\boldsymbol{\delta_t L(x_z, \omega)}$

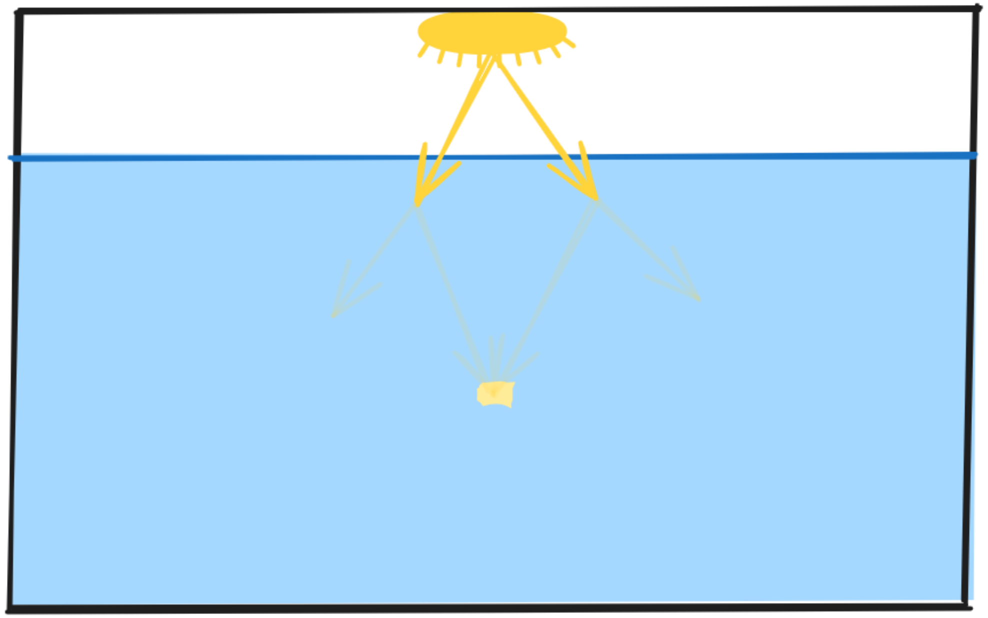
# Intuitive explanation of spectral rendering

Incoming Incident Light from all directions :

$$\frac{d L_{in-scattering}(x,\omega)}{dz} = \delta_s L_i(x,\omega)$$

Where: $L_i(x,\omega) = \int_\Omega L(x,\omega_i) f_s(x,\omega_i,\omega) d\omega_i$

$f_s(x,\omega_i,\omega)$ – Phase Function the likely-hood of scattering towards $\omega$



Light in-scattering

# Intuitive explanation of spectral rendering

The **RTE (Radiative Transfert Equation)** represent the amount of radiance gained, due to in-scattering, light coming from any direction scattered towards ω, minus the lost: out-scattered, or absorbed.

Combining both extinction and in-scattering : $\frac{dL(x_z,\omega)}{dz} = \frac{dL_{extinction}(x,\omega)}{dz} + \frac{dL_{in-scattering}(x,\omega)}{dz}$

**Radiative Transfer Equation :** $\frac{dL(x_z,\omega)}{dz} = -\delta_t L(x_z,\omega) + \int_\Omega L(x,\omega_i) f_s(x,\omega_i,\omega) d\omega_i$

# Intuitive explanation of spectral rendering

The **VRE (Volume rendering equation)** is the combination of the RTE for water plus the medium contribution on the ray. For water, the medium contribution is represented as how much radiance is lost when travelling from the objects to the camera.

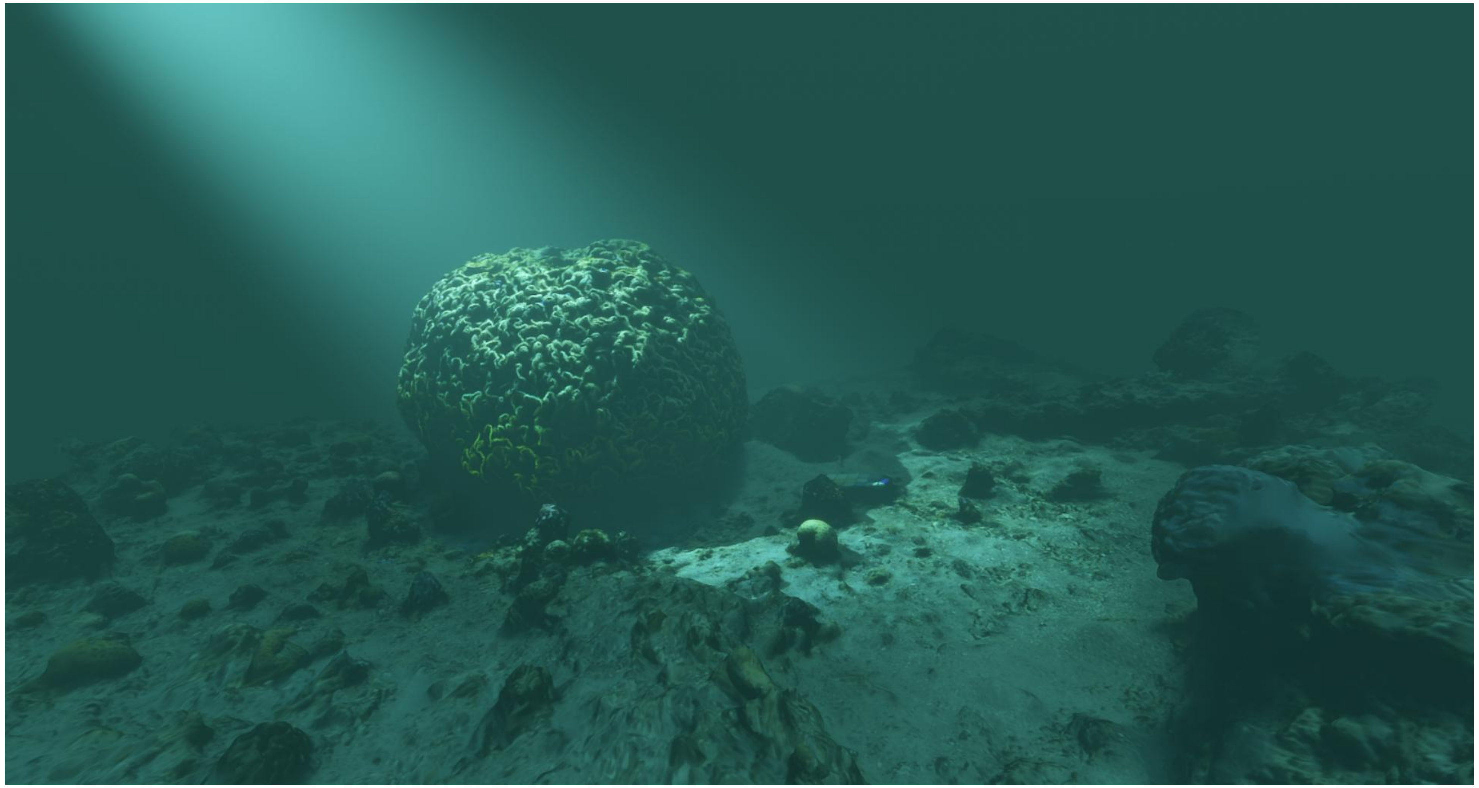
**Volume rendering equation**: $\boldsymbol{L(x_Z, \omega) = T(x_Z) L_s(x, \omega) + \int_{z=0}^{Z} T(x_z) L_i(x_z, \omega) dz}$

$$\boldsymbol{L_i = \int_{\Omega} L(x, \omega_i) f_s(x, \omega_i, \omega) d\omega_i}$$

$\boldsymbol{L_s}$ - Light at the surface (or exit) point $x_Z$.
$\boldsymbol{T(x_Z) = e^{-\sigma_t Z}}$ - Transmittance of water over distance Z

# Limitations of this approach

# Intuitive Limitations of this approach



Rendering with monte-carlo ray-tracing : **4 hours**


Sources : Real-Time Underwater Spectral Rendering, February 2024, Nestor Monzon;

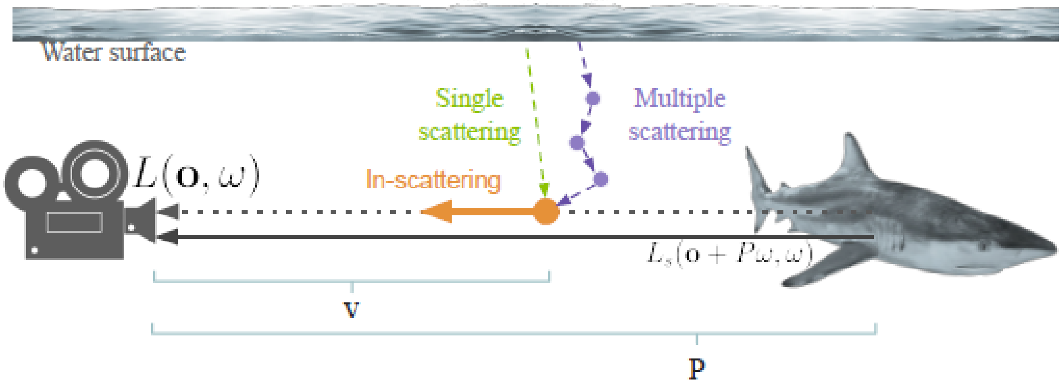
# Paper Approach

# Single and Multiple Scattering

*VLE:*

$$L(x_Z, \omega) = T(x_Z) L_s(x, \omega) + L_m$$

$$L_m = \int_{z=0}^{Z} T(x_z) L_i(x_z, \omega) dz$$



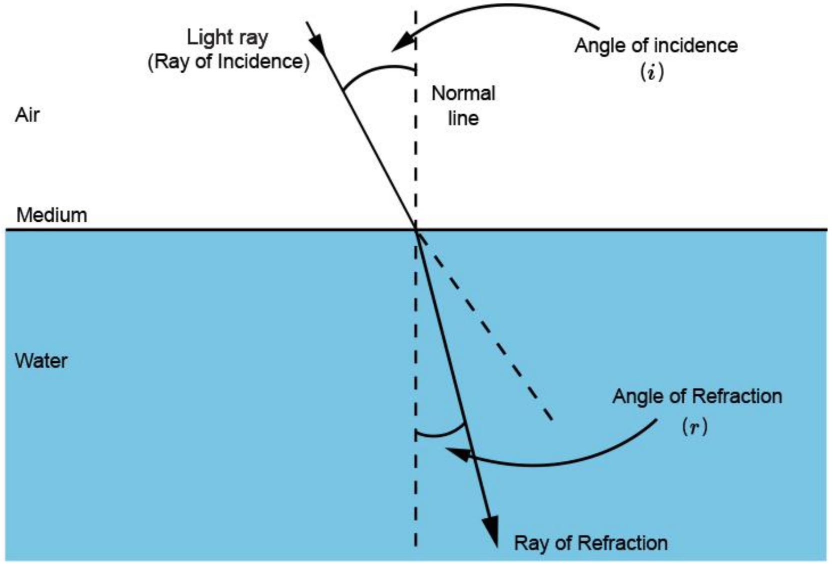

Light reaching Camera = Single Scattering + Multiple Scattering.

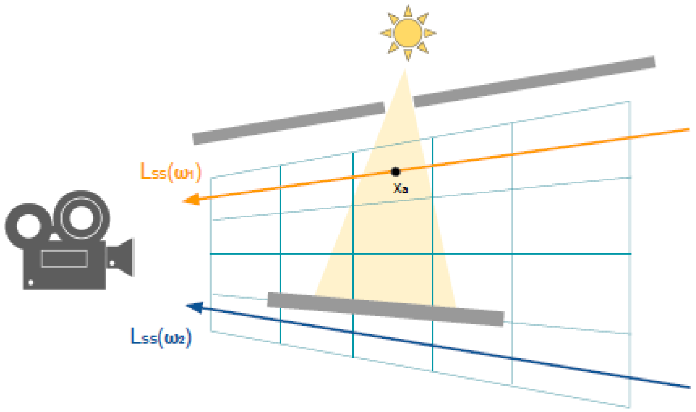
$$L_m(x_z,\omega) = \int_{z=0}^{Z} T(x_z) L_{iss}(x_z, \omega) dz + \int_{z=0}^{Z} T(x_z) L_{ims}(x_z, \omega) dz = L_{ss}(x_z, \omega) + L_{Ms}(x_z, \omega)$$

# Single Scattering

1. Compute refracted sunlight direction
2. Compute how sunlight attenuates with depth $\rightarrow e^{-y\cos(\theta_{sun})\delta_t}$
3. Build the froxel grid
4. Precompute single-scattered in-scattering in each froxel
5. Render pass begins — march rays through froxels
6. Accumulate the froxel contributions
7. Compute the Single-Scattering Irradiance:

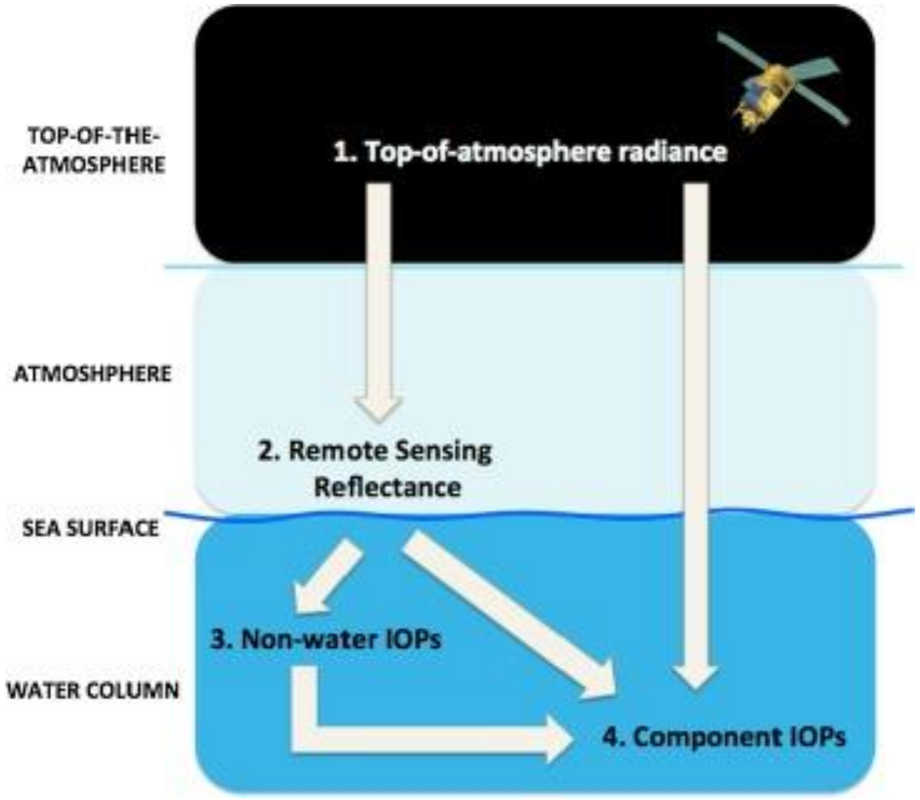
$$L_{ss} = \int_{z=0}^{Z} \boldsymbol{T(x_z)L_{iss}(x_z,\omega)dz} = d \sum_{x_z+\varepsilon\omega} \boldsymbol{T(x_z)L_{iss}(x_z+\varepsilon\omega,\omega)}$$



# Inherent and Apparent Optical Properties

**IOPs** – Water Defining behavior (Scattering and Absorption Coefficient)
**AOPs** – Underwater Optical Quantities that depend on Illumination Geometry and Camera Angle. (Well studied and Easier to measure)

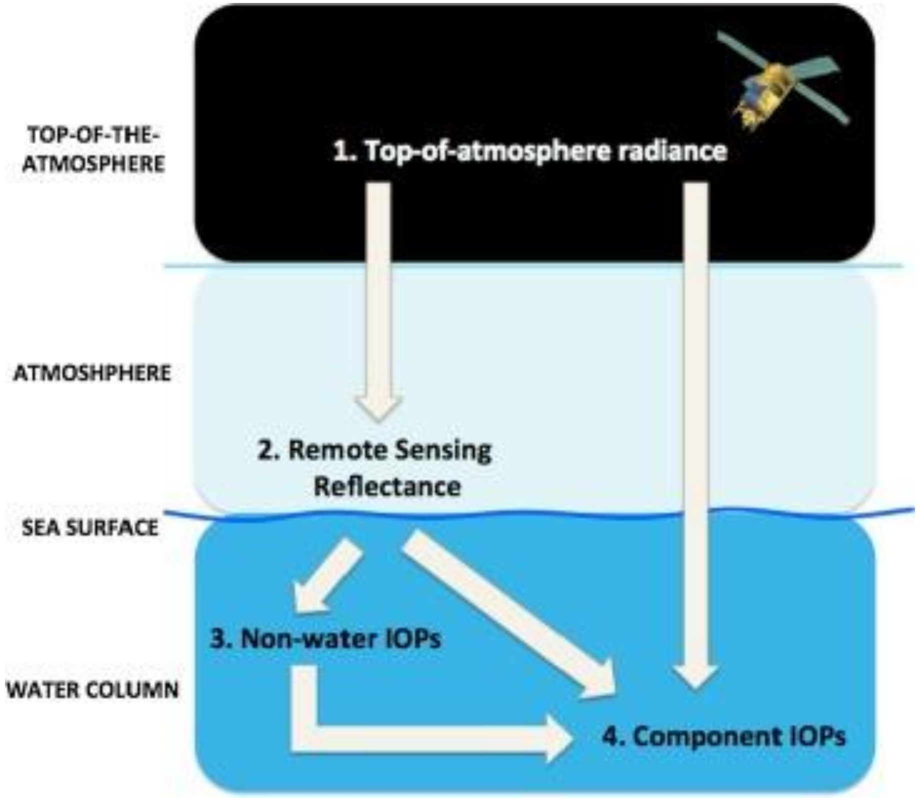


# Diffuse Downwelling attenuation $K_d$

KOP function (quasi-inherent optical property)
Diffuse Downwelling attenuation:

$$K_d = \frac{1}{E(y)} \frac{dE(y)}{dy}$$

Y – Depth; E(Y) – Radiance at depth y.
Describes how quickly light decreases with depth after multiple scattering.
Different Water Types, defined differently



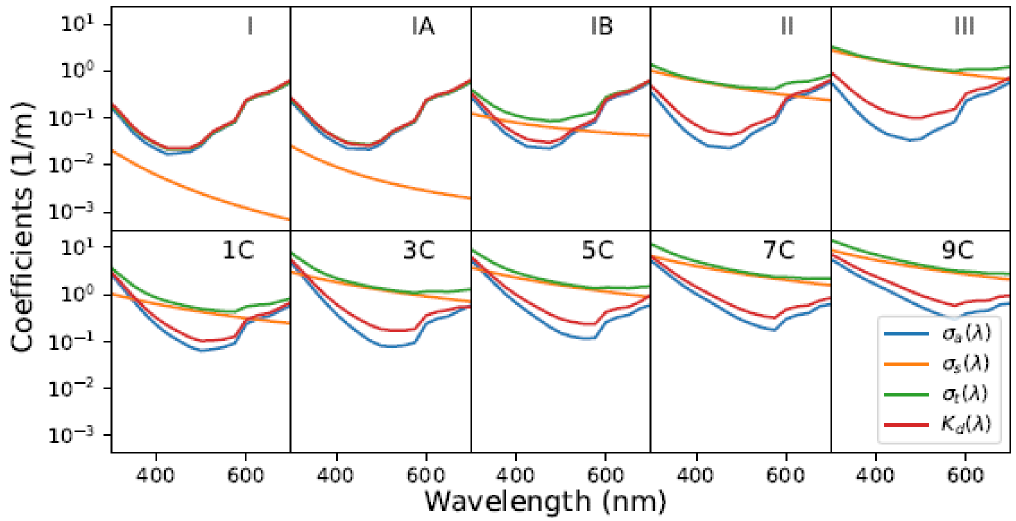
# Asymptotic Radiance Distribution

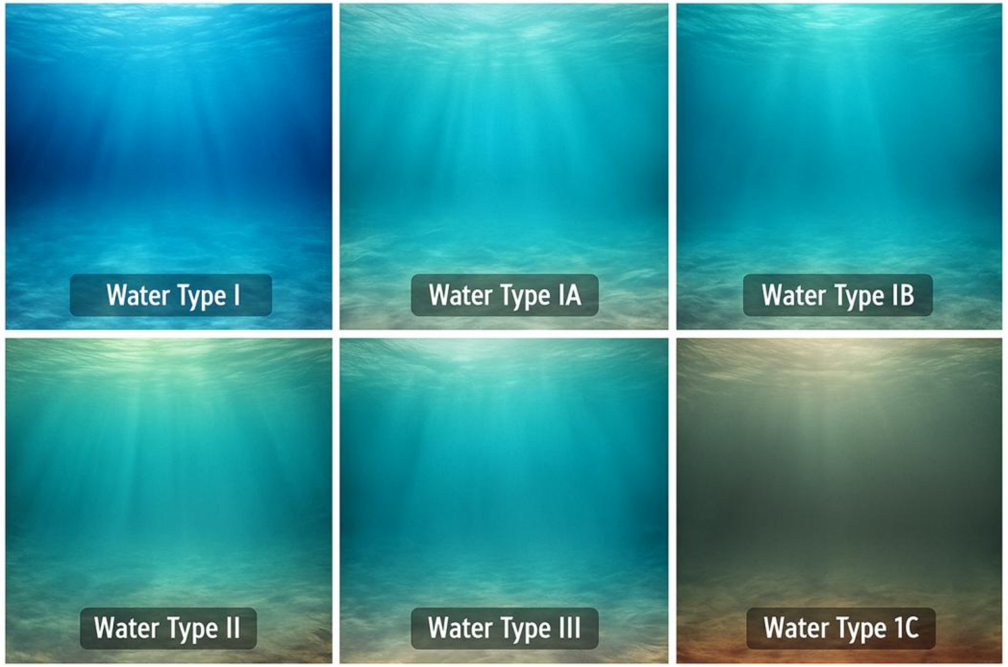
The surface level of the water, shows a higher dependence on light geometric and angles.
As we go deeper into the water, the lighting becomes homogeneous and light geometry-independent:

$$\text{AOPs} \rightarrow \text{IOPs}$$

$$\text{K}_\text{d} \rightarrow \text{Dependent on } \sigma_\text{a} \text{ and } \sigma_\text{s}$$

# Jerlov Water Types



# Depth-Dependent Radiance

Assumption: Spatially Homogeneous Medium.

The Downwelling Radiance $E_D$ in the Medium depends only on E(0):

$$E_D(y) = E_D(0)e^{-K_d y}$$

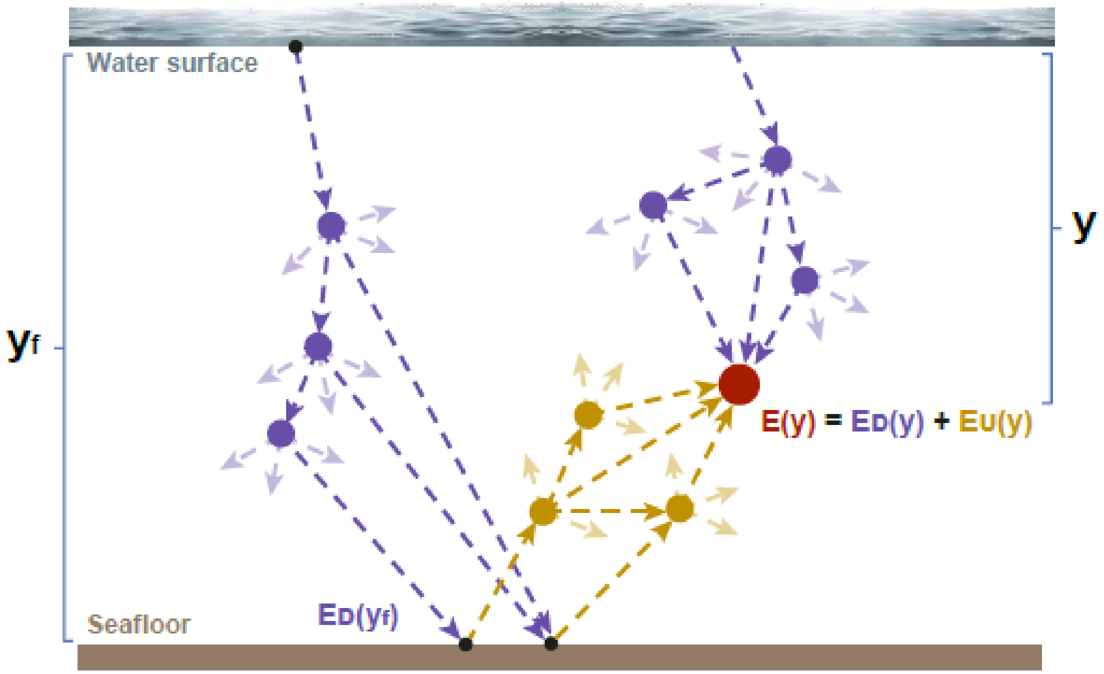
As we go deeper, the light from the surface decreases exponentially.

Assumption: Lambertian horizontal Floor

The Upwelling Radiance $E_U$ in the Medium depends on the $E_D$ and E(0).

$$E_U(y_f) = \frac{r}{\pi} E_D(y_f) = \frac{r}{\pi} E_D(0)e^{-K_d y}$$

For reflectance coefficient r



# Multiple Scattering

Sum the Upwelling and Downwelling radiance:

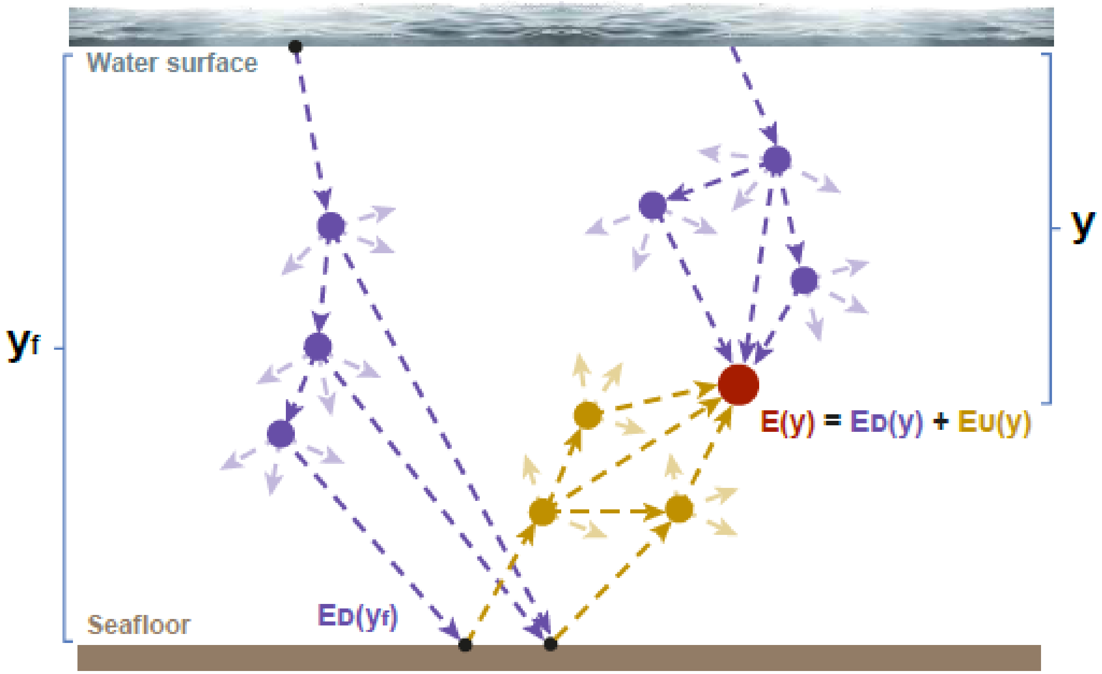
$$E(y) = E_D(y) + E_U(y_f)e^{-|y-y_f|K_d} = E(0)\left(e^{-K_d y} + \frac{r}{\pi}e^{-(2y_f-y)K_d}\right)$$

Multiple Scattering Radiance:

$$L_{Ms}(x_z,\omega) = \int_{z=0}^{Z} T(\varepsilon)\delta_s f_s(x_z + \omega\varepsilon, \omega_i, \omega)E(y)dz$$

Assumption: isotropic phase function; $f_s = \frac{1}{4\pi}$

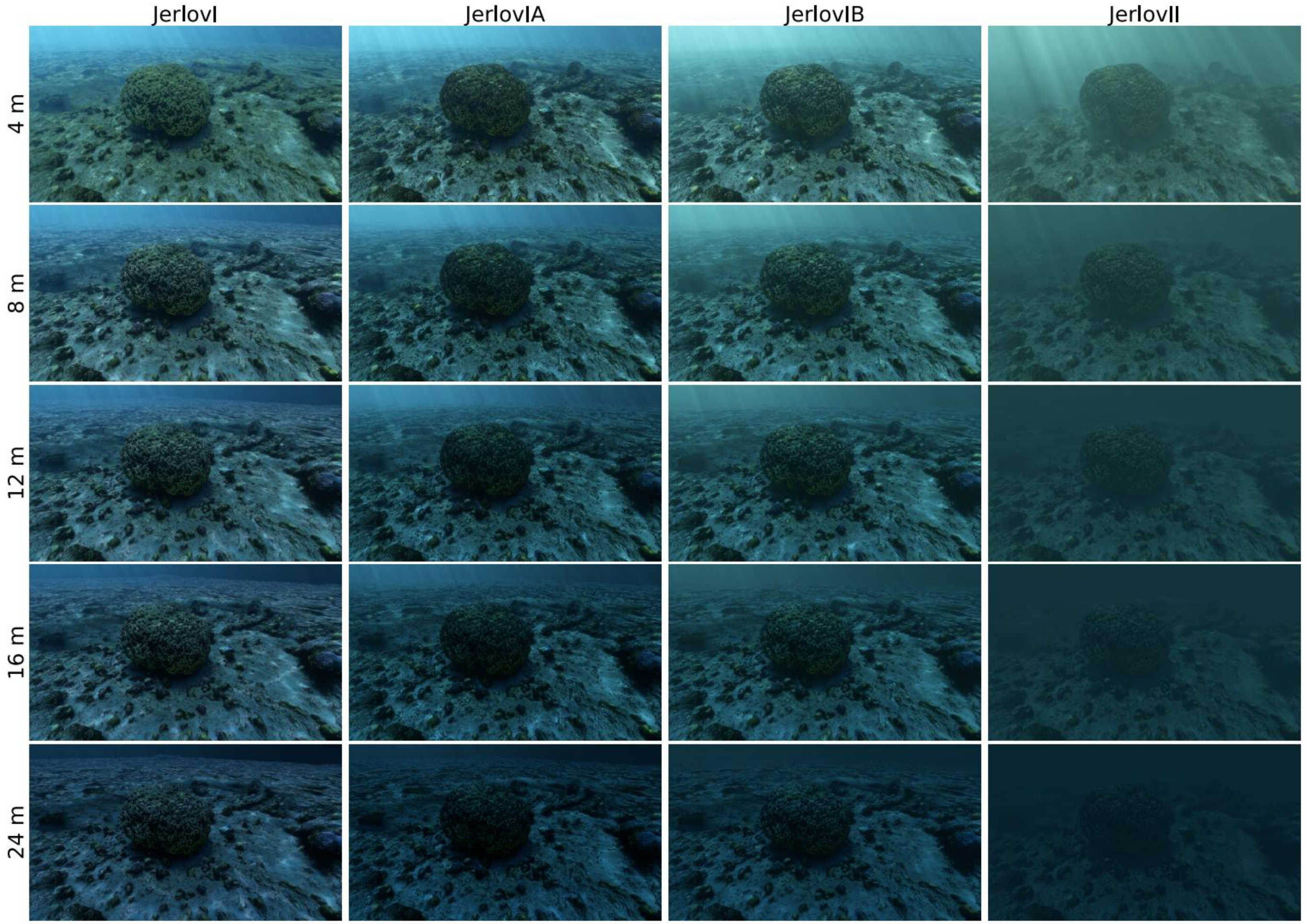
$$L_{Ms}(x_z,\omega) = \frac{\delta_s E_d(0)}{4\pi(K_d y_\omega - \delta_t)}(e^{(K_d y_\omega - \delta_t)Z} - 1)(e^{-K_d y_{x_z}} + \frac{r}{\pi}e^{-K_d(2y_f - y_{x_z})})$$




Idea: Simplify the Computation of multiple scattering into a single O(1) expression of constant computational time (per wavelength, per pixel).

# Results

# Results

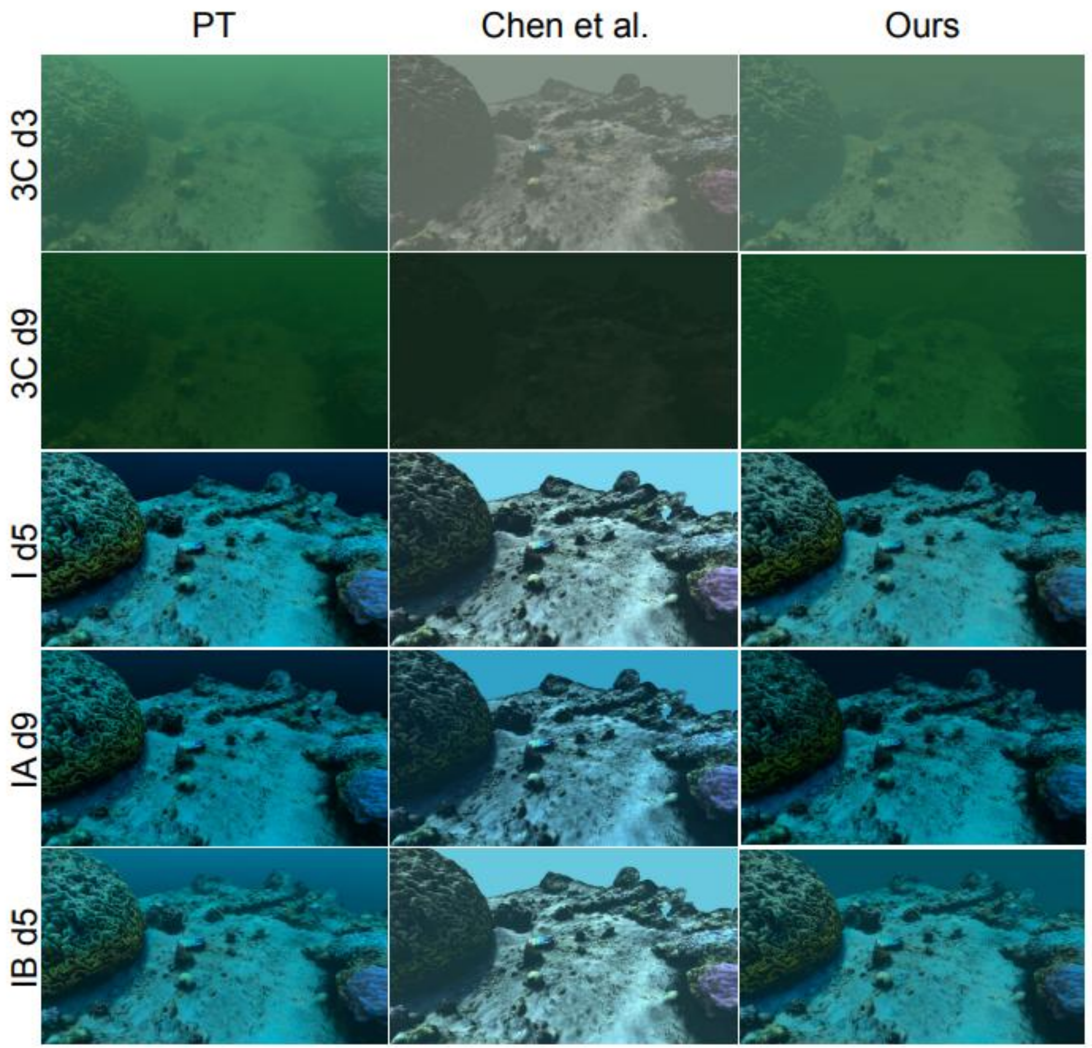



Paper results for multiple water types and depths



Sources : Real-Time Underwater Spectral Rendering, February 2024, Nestor Monzon;

# Results




Comparison with ground truth and state of the art method


Sources : Real-Time Underwater Spectral Rendering, February 2024, Nestor Monzon;

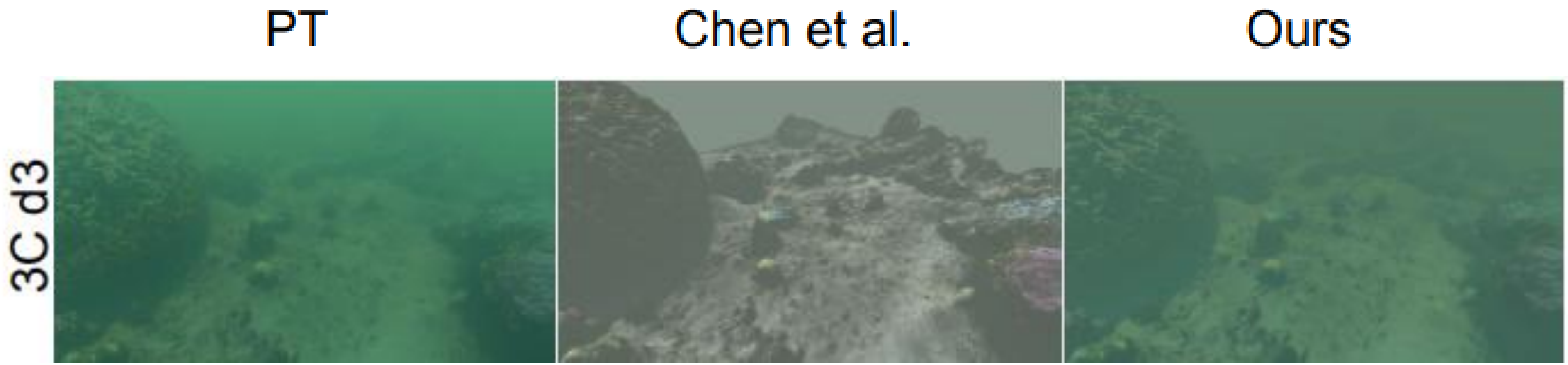
# Results


State of the art (Chen et al.) :

- Overall blue color is not the right tint
- Background blue color is much brighter


Sources : Real-Time Underwater Spectral Rendering, February 2024, Nestor Monzon;

# Results

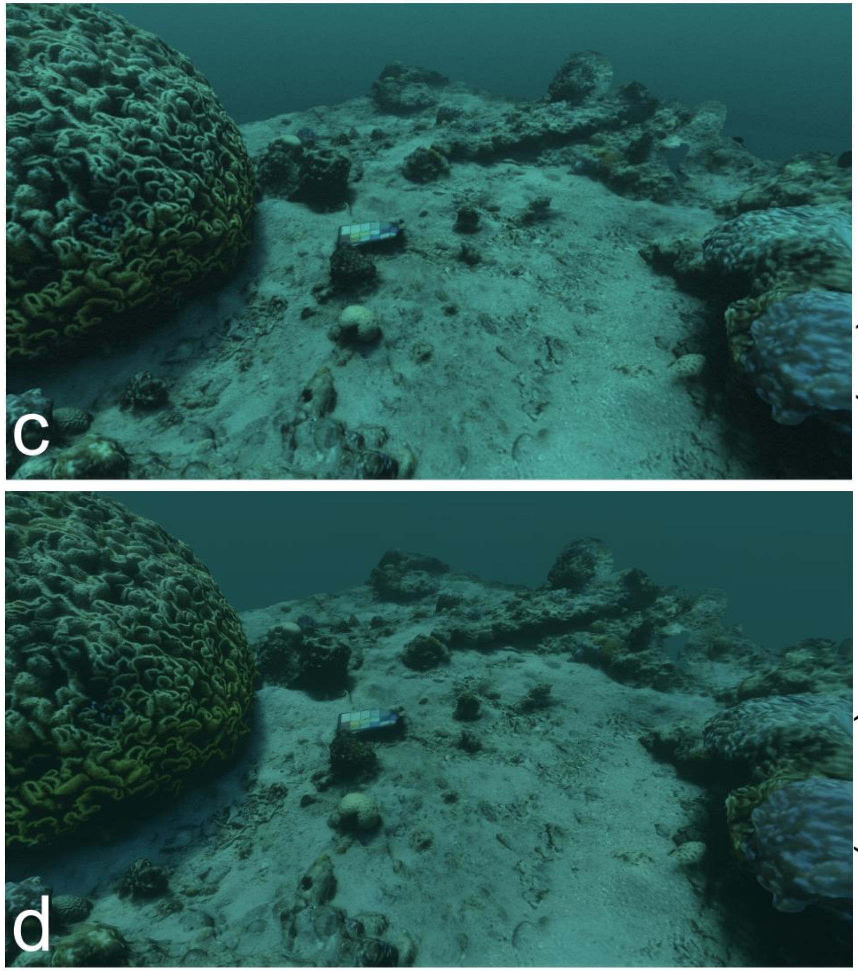



State of the art (Chen et al.) :

- Water looks gray
- Depths of field blur is not present

 Sources : Real-Time Underwater Spectral Rendering, February 2024, Nestor Monzon; 

# Results




Ground truth (path tracer)
4 hours

Paper method
30 ms

Rendering time comparison between paper method and ground truth


Sources : Real-Time Underwater Spectral Rendering, February 2024, Nestor Monzon;

# Thank you

Group 1
Course: CS580
Speakers: Daniel Mocanu, Valentin Le Lièvre

# Quiz