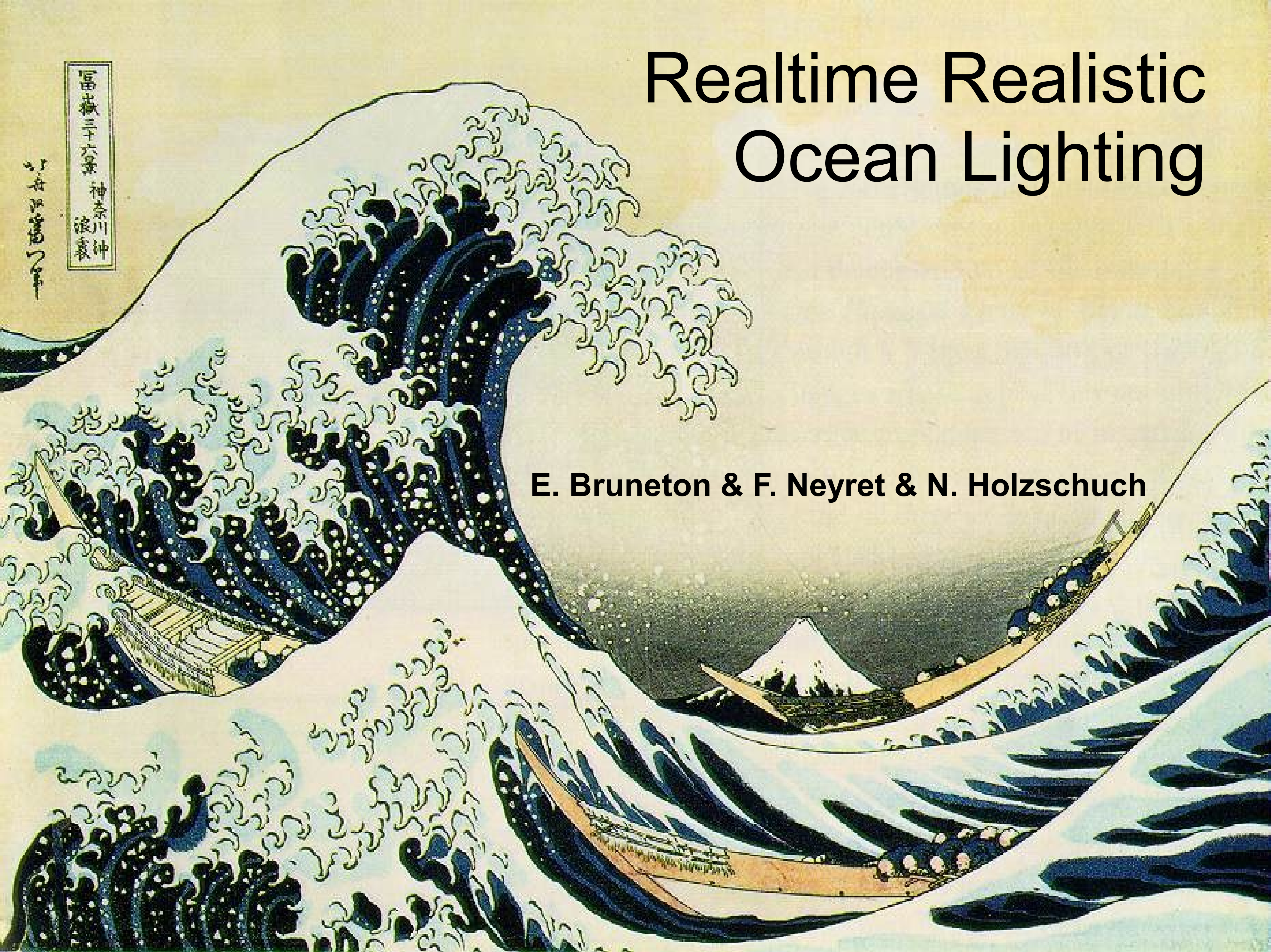
Realtime Realistic Ocean Lighting

E. Bruneton & F. Neyret & N. Holzschuch

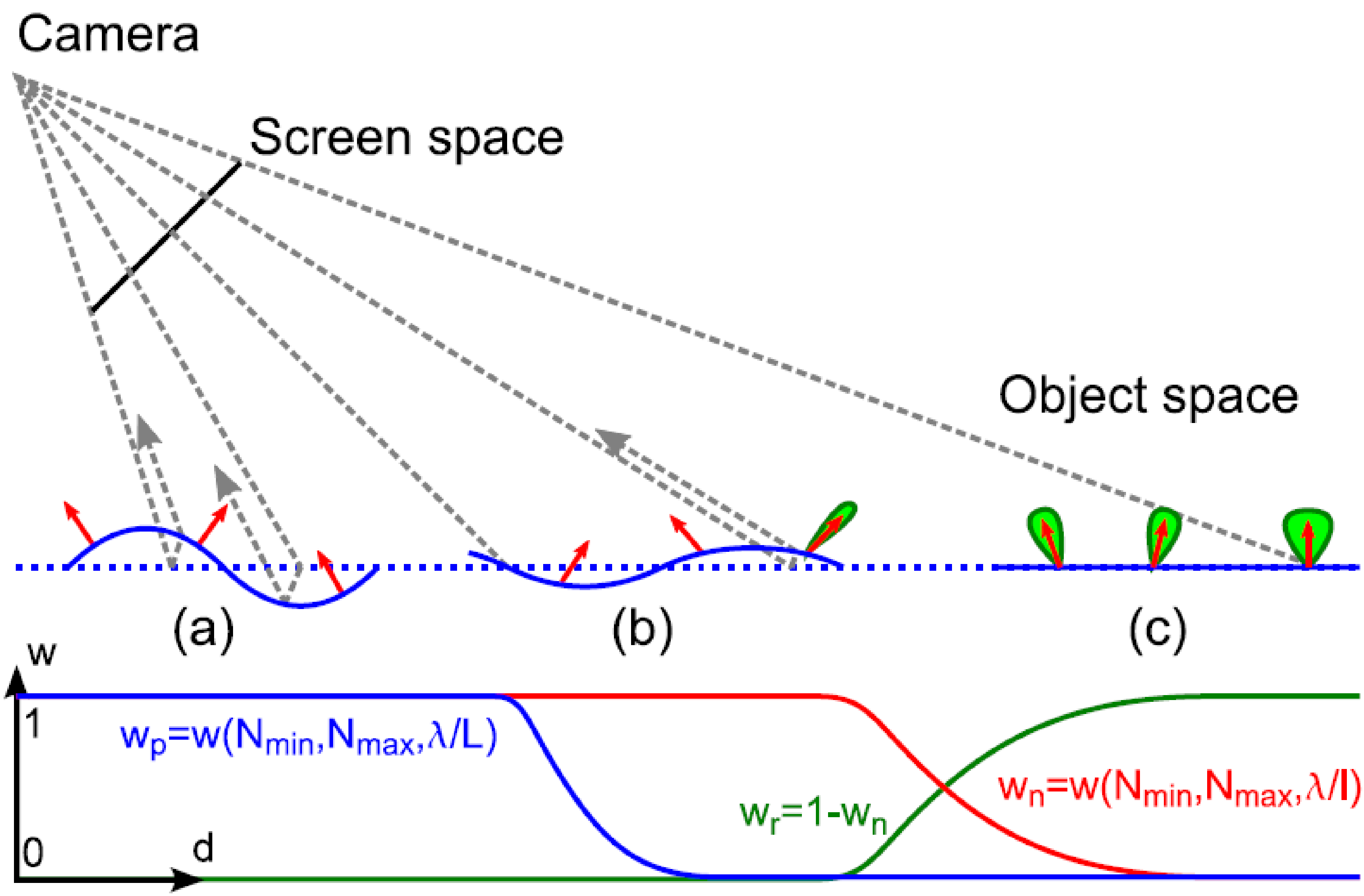


Motivation

- ocean surface is a highly complex problem
- storage expensive: waves at different wavelengths
- rendering expensive: waves at all distances
- illumination: sun, environment map, underwater scattering
- dynamic - no precomputations

Real-time Rendering

- Hierarchical approach: Geometry - Normals - BRDF



Hierarchical approach

- a regular grid in screen space is projected on the horizontal plane

- points are displaced by waves and projected back
  -> creates geometry

- compute normals per pixel
- use BRDF per point
  -> used for shading

Transition between details

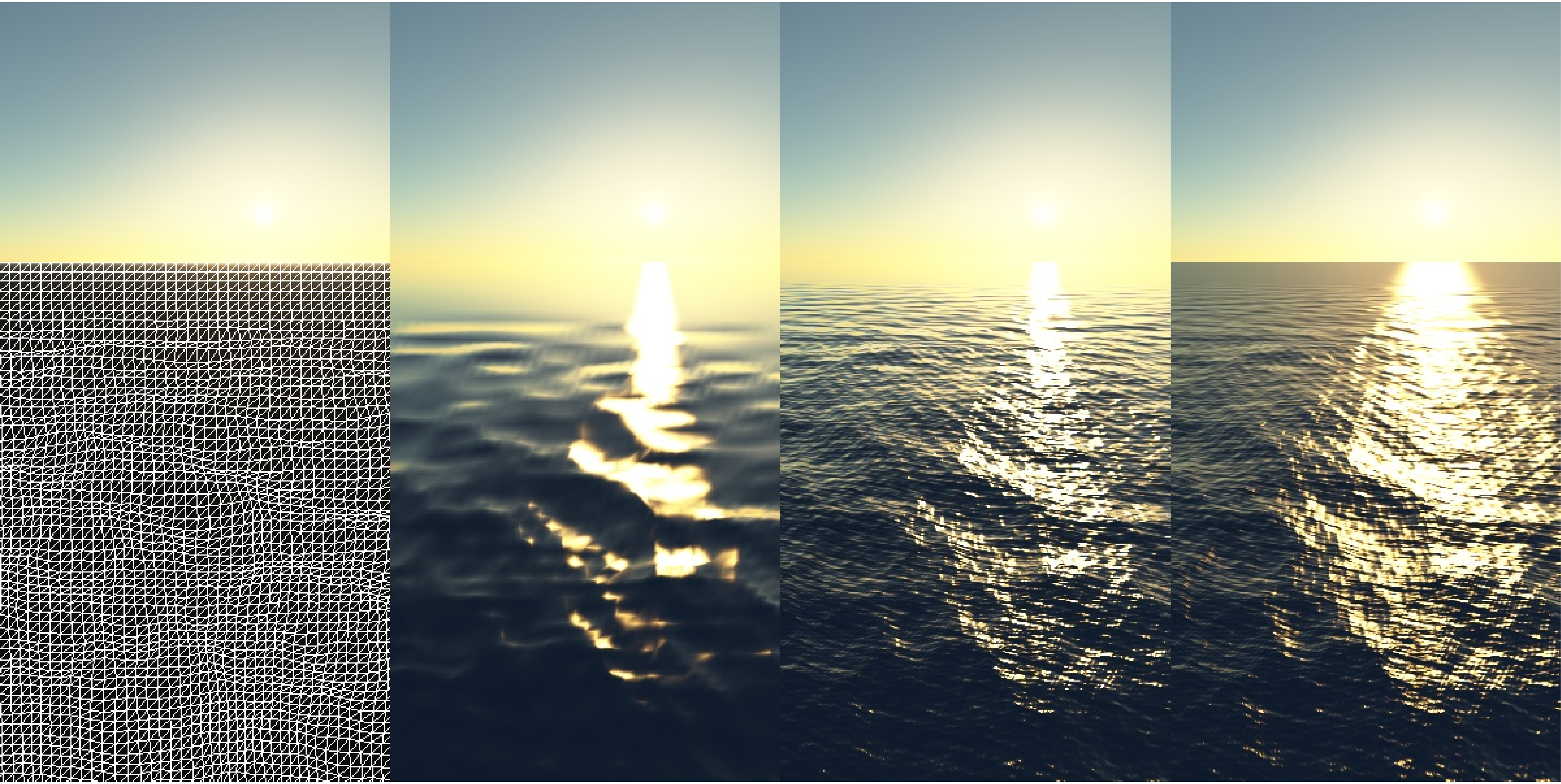
- at each level, the waves are attenuated by factor according to their wavelength

- grid points - wavelengths > grid size
- normals - wavelengths > pixel size
- BRDF - the rest

Attenuation parameter:

$w(a, b, x) = 3x'^2 - 2x'^3$

$x' = clamp((x-a)/(b-a), 0, 1)$

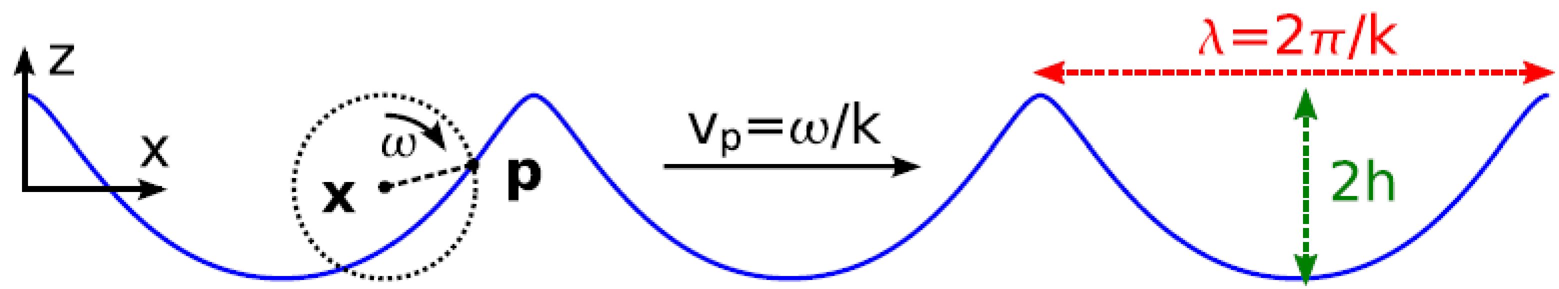


Geometry model

- deep ocean waves
- geometry: sum of trochoids
- hierarchical representation

$\mathbf{p} = [\mathbf{x} + h \sin(\omega t - kx), h \cos(\omega t - kx)]$

$\omega = \sqrt{gk}$



- N trochoids picked from spectrum

Close distance waves

- Covers wavelengths greater than λ / L (L - the size of the projected grid cell)

- Gerstner waves: each point is dispaced according to:

$$\mathbf{p} = \begin{bmatrix} \mathbf{x} \\ 0 \end{bmatrix} + \sum_{1}^{n} w_{p,i}\, \mathbf{t}_i, \quad \mathbf{t}_i = \begin{bmatrix} \frac{\mathbf{k}_i}{\|\mathbf{k}_i\|} h_i \sin(\omega_i t - \mathbf{k}_i \cdot \mathbf{x}) \\ h_i \cos(\omega_i t - \mathbf{k}_i \cdot \mathbf{x}) \end{bmatrix}$$

Attenuation parameter:

w_p = w(Nmin, Nmax, λ/L)

Normals

- average normal inside a pixel
- we clamp subpixel waves

$$\mathbf{n} = \left(\begin{bmatrix} \frac{\partial \mathbf{x}}{\partial x} \\ 0 \end{bmatrix} + \sum_{1}^{n} w_{n,i} \frac{\partial \mathbf{t}_i}{\partial x} \right) \wedge \left(\begin{bmatrix} \frac{\partial \mathbf{x}}{\partial y} \\ 0 \end{bmatrix} + \sum_{1}^{n} w_{n,i} \frac{\partial \mathbf{t}_i}{\partial y} \right)$$

Attenuation parameter:

w_p = w(Nmin, Nmax, λ/l)

[l - size of a projected pixel]

BRDF

- subpixel surface details
- microfacet model
- trochoids - independent random variables
- CLT - sum of trochoids: surface with slopes with Gaussian distribution with variance:

$$\begin{bmatrix}\sigma_x^2\\ \sigma_y^2\end{bmatrix} = \sum_1^n \frac{[k_{i,x}^2 \;\; k_{i,y}^2]^T}{\|\mathbf{k}_i\|^2}\left(1-\sqrt{1-\|\mathbf{k}_i\|^2 w_r^2 h_i^2}\right)$$

Attenuation: w_r = 1 - w_n
[1 - normal attenuation]

Ocean BRDF

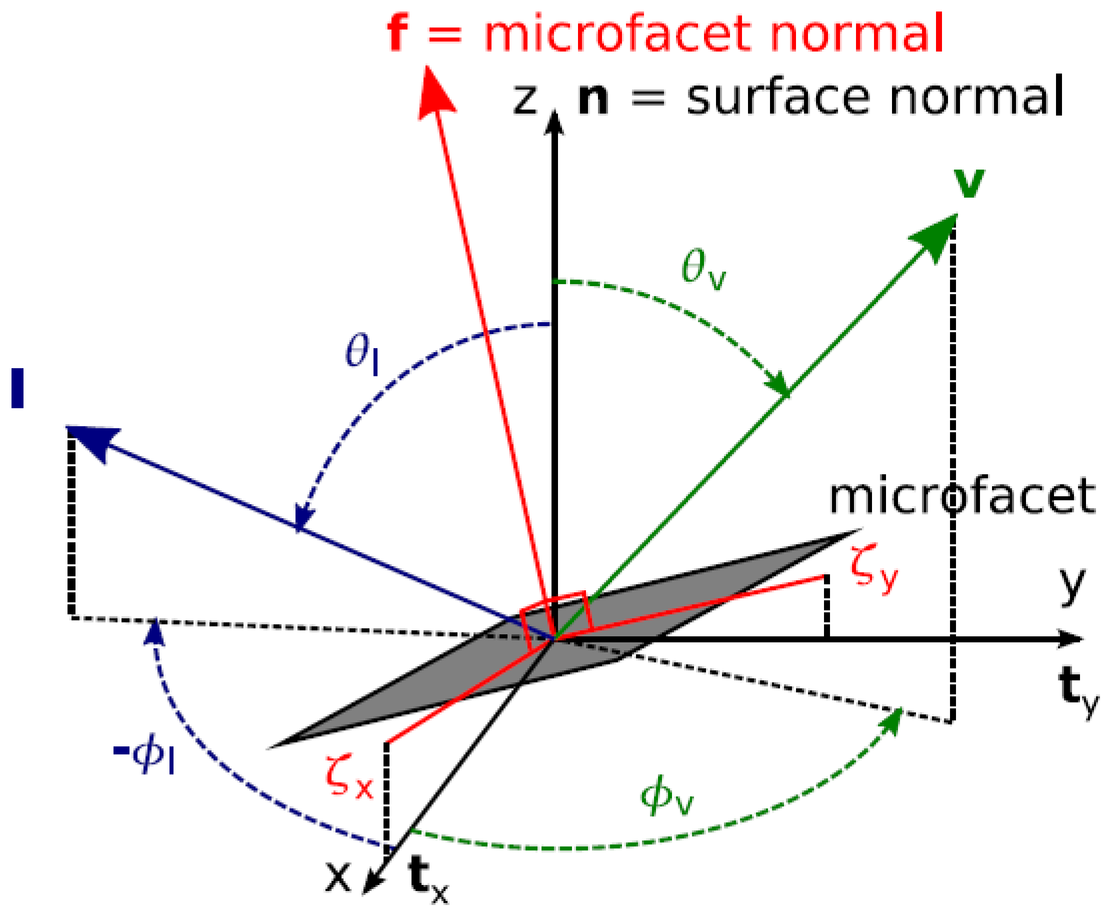


Figure 5: ***BRDF model coordinates*** *(from [RDP05]).* **v** *and* **l** *are unit vectors towards the viewer and the light.* **f** *is the normal of a microfacet whose x and y slopes are ζ_x and ζ_y.*

Ocean BRDF

$$q_{vn}(\boldsymbol{\zeta}, \mathbf{v}, \mathbf{l}) = \frac{p(\boldsymbol{\zeta}) \max(\mathbf{v} \cdot \mathbf{f}, 0) H(\mathbf{l} \cdot \mathbf{f})}{(1 + \Lambda(a_v) + \Lambda(a_l)) f_z \cos\theta_v} \mathrm{d}^2 \boldsymbol{\zeta}$$

$$\mathbf{f}(\boldsymbol{\zeta}) = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \frac{1}{\sqrt{1 + \zeta_x^2 + \zeta_y^2}} \begin{bmatrix} -\zeta_x \\ -\zeta_y \\ 1 \end{bmatrix}$$

$$p(\boldsymbol{\zeta}) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{1}{2}\left(\frac{\zeta_x^2}{\sigma_x^2} + \frac{\zeta_y^2}{\sigma_y^2}\right)\right)$$

$$\Lambda(a_i) = \frac{\exp(-a_i^2) - a_i\sqrt{\pi}\,\mathrm{erfc}(a_i)}{2a_i\sqrt{\pi}}, i \in \{v, l\}$$

$$a_i = \left(2\left(\sigma_x^2 \cos^2\phi_i + \sigma_y^2 \sin^2\phi_i\right) \tan\theta_i\right)^{-1/2}$$

$$q_{vn}^e(\boldsymbol{\zeta}, \mathbf{v}) = \frac{p(\boldsymbol{\zeta}) \max(\mathbf{v} \cdot \mathbf{f}, 0)}{(1 + \Lambda(a_v)) f_z \cos\theta_v} \mathrm{d}^2 \boldsymbol{\zeta}$$

$$\iint_{-\infty}^{\infty} q_{vn}^e(\boldsymbol{\zeta}, \mathbf{v}) \mathrm{d}^2 \boldsymbol{\zeta} = 1$$

Ocean BRDF

Change the integral measure from slopes to directions:

$$\mathrm{d}^2\boldsymbol{\zeta} = \frac{\sin\theta_l \,\mathrm{d}\theta_l \,\mathrm{d}\phi_l}{4h_z^3 \mathbf{v}\cdot\mathbf{h}} = \frac{\mathrm{d}^2\boldsymbol{\omega}_l}{4h_z^3 \mathbf{v}\cdot\mathbf{h}}$$

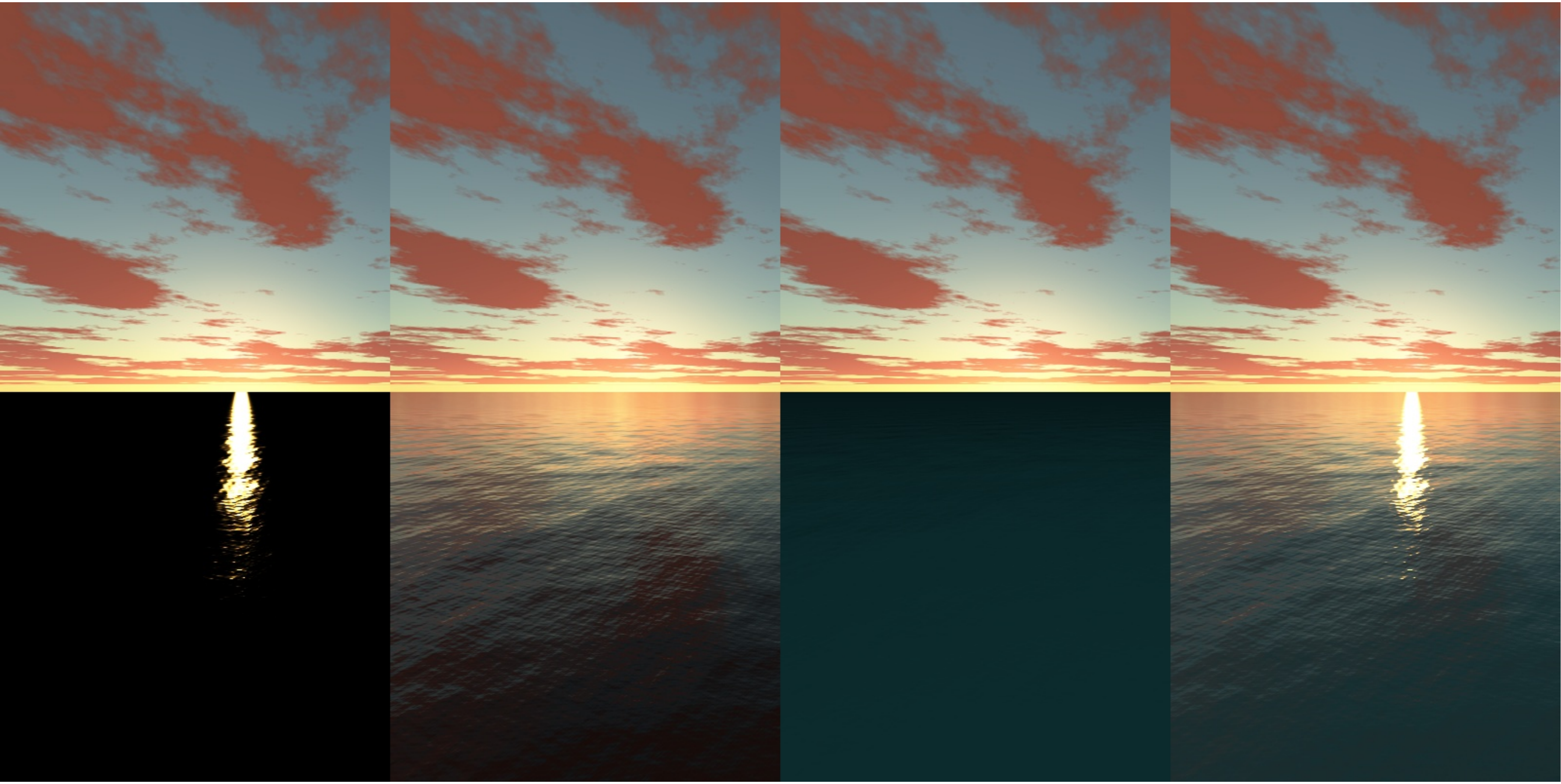
$$\mathrm{brdf}(\mathbf{v},\mathbf{l}) = \frac{q_{vn}(\boldsymbol{\zeta}_h,\mathbf{v},\mathbf{l})F(\mathbf{v}\cdot\mathbf{h})}{4h_z^3 \cos\theta_l \mathbf{v}\cdot\mathbf{h}}$$

Work flow

- Project the points of the grid onto the plane, displace, and project back
- Compute normals per pixel
- Compute the lighting [is going to be described now]

Rendering

- Sun light
- Sky light
- Refracted light



Sun light

- sun reflected at point **p:**
  - apply BRDF in the tangent plane allinged with the average normal **n** and the wind direction

- slope variances modelled by the Gauss distribution

$$I_{sun} \approx L_{sun}\Omega_{sun}\, p(\boldsymbol{\zeta}_h)\frac{R+(1-R)(1-\mathbf{v}\cdot\mathbf{h})^5}{4h_z^4\cos\theta_v(1+\Lambda(a_v)+\Lambda(a_l))}$$

Sky light

- microfacet model
  brdf (v, l) = p(ς) ρ(v, l)

- illumination from sky:

$$I_{sky} = \iint_{\Omega} p(\boldsymbol{\zeta}_h) \rho(\mathbf{v}, \mathbf{l}) L_{sky}(\mathbf{l}) \cos\theta_l \, \mathrm{d}^2\boldsymbol{\omega}_l$$

$$I_{sky} = \iint_{-\infty}^{\infty} p(\boldsymbol{\zeta}) \rho'(\mathbf{v}, \boldsymbol{\zeta}) L_{sky}(\mathbf{r}) H(r_z) \mathrm{d}^2\boldsymbol{\zeta}$$

$$I_{sky} \approx \bar{F}\bar{L}, \quad \bar{F}(\mathbf{v}) = \iint_{-\infty}^{\infty} p(\boldsymbol{\zeta}) \rho'(\mathbf{v}, \boldsymbol{\zeta}) H(r_z) \mathrm{d}^2\boldsymbol{\zeta}$$

$$\bar{L}(\mathbf{v}) = \iint_{-\infty}^{\infty} p(\boldsymbol{\zeta}) L_{sky}(\mathbf{r}) H(r_z) \mathrm{d}^2\boldsymbol{\zeta}$$

Average Fresnel reflectance

$$\bar{F}(\mathbf{v}) \approx R + (1-R)\iint_{-\infty}^{\infty} q^{e}_{vn}(\boldsymbol{\zeta}, \mathbf{v})(1-\mathbf{v}\cdot\mathbf{h})^{5}\,\mathrm{d}^{2}\boldsymbol{\zeta}$$

$$\sigma_v^2 = \sigma_x^2 \cos^2\phi_v + \sigma_y^2 \sin^2\phi_v$$

Approximation. Fitting function:

$$\bar{F}(\mathbf{v}) \approx R + (1-R)\frac{(1-\cos\theta_v)^{5\exp(-2.69\sigma_v)}}{1+22.7\sigma_v^{1.5}}$$

Sky radiance

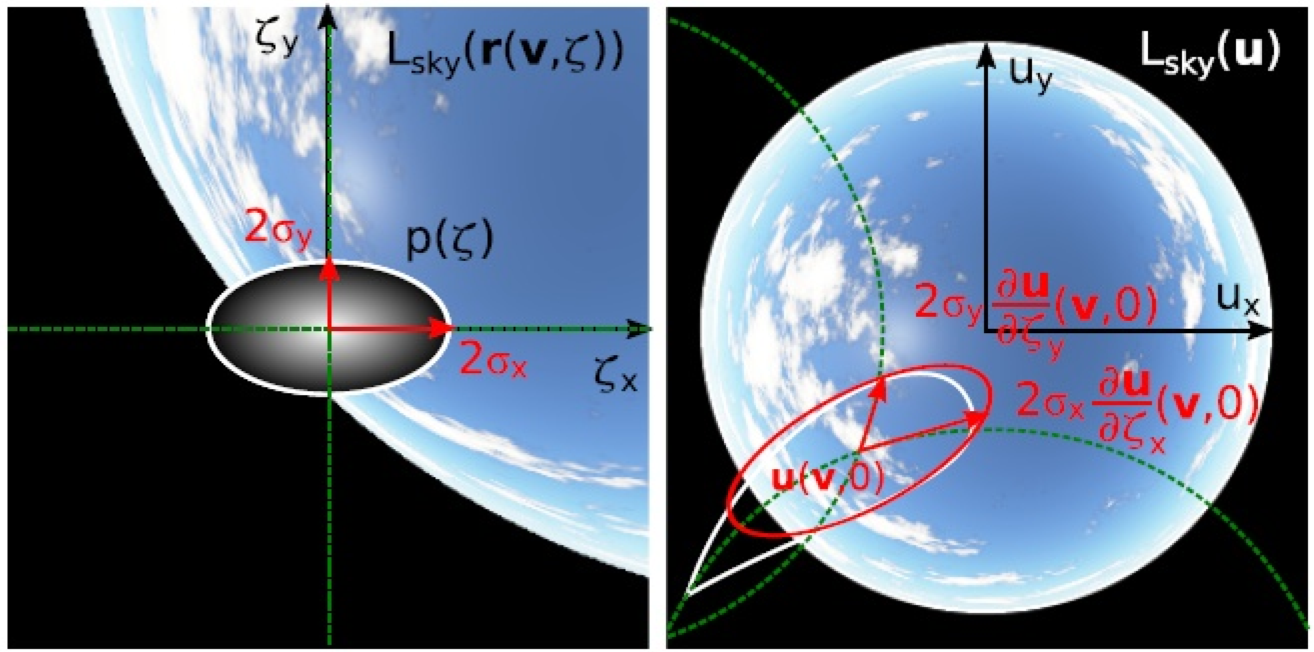
- environment map

- for better performance - filtering

- ellipse dependent on the Gaussian slope distributions

- parametrization: stereographic projection - so that we can filter with an ellipse around the mean reflectance

Map filtering



$$\bar{L} \approx \mathrm{tex2D}(\mathbb{L}, \mathbf{u}(\mathbf{v},0), 2\sigma_x \frac{\partial \mathbf{u}}{\partial \zeta_x}(\mathbf{v},0), 2\sigma_y \frac{\partial \mathbf{u}}{\partial \zeta_y}(\mathbf{v},0))$$

Refracted light

• the radiance reaching the surface from below - considered diffuse (because of multiple scattering)

• propotional to the sun and sky irradiance

•Replace BRDF with:
Transmittance = 1 - Reflectance

$$I_{sea} \approx L_{sea}(1 - \bar{F})$$

Algorithm

Algorithm 5.1: $\textsc{SeaColor}(\mathbf{v}, \mathbf{l}, \mathbf{n}, \mathbf{t}_x, \mathbf{t}_y, \sigma_x, \sigma_y)$

procedure $\mathrm{U}(\zeta)$

$\mathbf{f} \leftarrow \mathrm{normalize}([-\zeta_x \;\; -\zeta_y \;\; 1])$ *// tangent space*

$\mathbf{f} \leftarrow f_x \mathbf{t}_x + f_y \mathbf{t}_y + f_z \mathbf{n}$ *// world space*

$\mathbf{r} \leftarrow 2(\mathbf{f} \cdot \mathbf{v})\mathbf{f} - \mathbf{v}$

return $[r_x \;\; r_y]/(1 + r_z)$

$\mathbf{h} \leftarrow \mathrm{normalize}(\mathbf{v} + \mathbf{l})$

$\zeta_h \leftarrow -[\mathbf{h} \cdot \mathbf{t}_x \;\; \mathbf{h} \cdot \mathbf{t}_y]/\mathbf{h} \cdot \mathbf{n}$

$\cos\theta_v \leftarrow \mathbf{v} \cdot \mathbf{n} \quad \phi_v \leftarrow \mathrm{atan}(\mathbf{v} \cdot \mathbf{t}_y, \mathbf{v} \cdot \mathbf{t}_x)$

$\cos\theta_l \leftarrow \mathbf{l} \cdot \mathbf{n} \quad \phi_l \leftarrow \mathrm{atan}(\mathbf{l} \cdot \mathbf{t}_y, \mathbf{l} \cdot \mathbf{t}_x)$

$\sigma_v \leftarrow (\sigma_x^2 \cos^2 \phi_v + \sigma_y^2 \sin^2 \phi_v)^{1/2}$

$\bar{F} \leftarrow R + (1 - R)(1 - \cos\theta_v)^{5e^{-2.69\sigma_v}} / (1 + 22.7\sigma_v^{1.5})$

$\mathbf{u}_0 \leftarrow \mathrm{U}([0 \;\; 0])$

$\Delta\mathbf{u}_x \leftarrow 2\sigma_x(\mathrm{U}([\varepsilon \;\; 0]) - \mathbf{u}_0)/\varepsilon$

$\Delta\mathbf{u}_y \leftarrow 2\sigma_y(\mathrm{U}([0 \;\; \varepsilon]) - \mathbf{u}_0)/\varepsilon$

$I_{sun} \leftarrow L_{sun}\Omega_{sun} \frac{p(\zeta_h)(R + (1-R)(1-\mathbf{v}\cdot\mathbf{h})^5)}{4(\mathbf{h}\cdot\mathbf{n})^4 \cos\theta_v (1 + \Lambda(a_v) + \Lambda(a_l))}$

$I_{sky} \leftarrow \bar{F}\, \mathrm{texture2DGrad}(L_{sky}, \mathbf{u}_0, \Delta\mathbf{u}_x, \Delta\mathbf{u}_y)$

$I_{sea} \leftarrow L_{sea}(1 - \bar{F})$

return $I_{sun} + I_{sky} + I_{sea}$

[video]

Thank you for your attention!