Advanced Computer Graphics
CS 563: *Real-Time Ocean Rendering*
[Real-Time Realistic Ocean Lighting using Seamless Transitions from Geometry to BRDF]

Xin Wang

March, 20, 2012

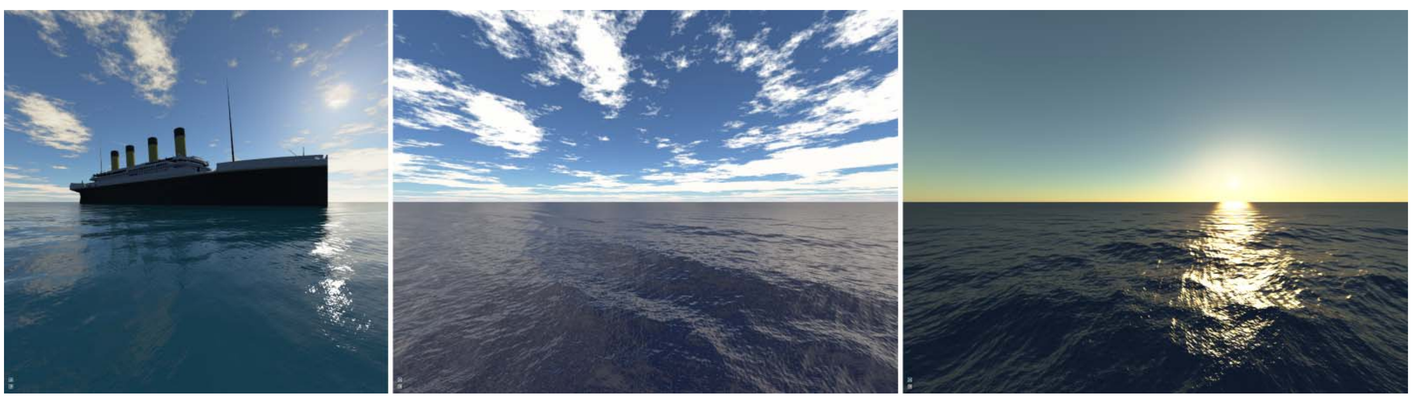
Computer Science Dept.

Worcester Polytechnic Institute (WPI)

Background

- Photorealistic rendering image
  - Cannot be used in games, simulators, etc…

- Realistic animation and rendering



Introduction

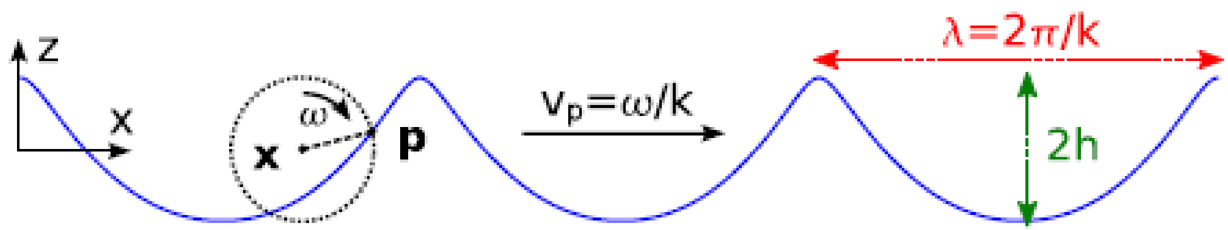
- Hierarchical modeling of the ocean
- Illumination reflection using BRDF
  - Lighting effects
  - BRDF model
- Approximate formula for computing the surfaces
- Rendering

Pervious Work

- Physical ocean models.
  - [CM54,PM64,RD07]
- Computer graphics ocean models.
  - [Tes01,CC06,HVT*06]
- Reflectance models.
  - [CT81,AS00,RDP05]
- Multi-resolution reflectance models.
  - [Kaj85,HSR07]

Ocean Model – Phase I

- Dynamic scene, no pre-computations
- physical facts about **deep water waves**

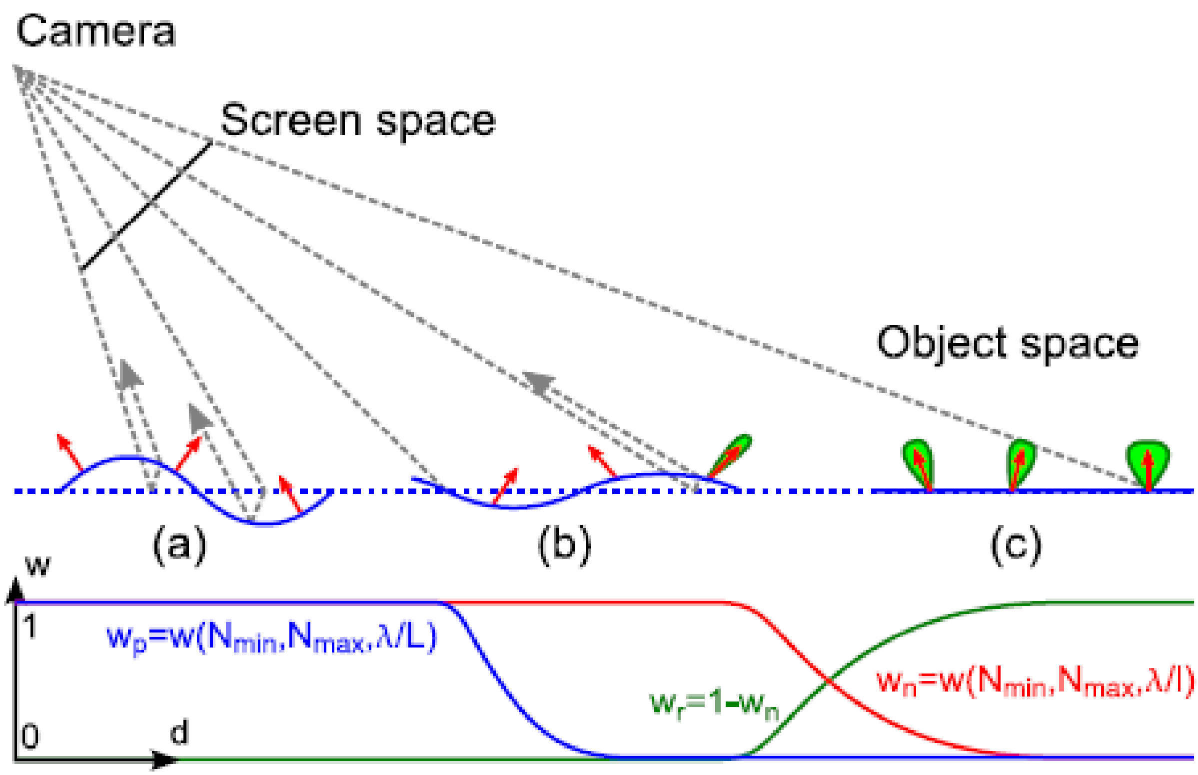


- Trochoid Waves.
  - A gerstner wave is defined by

p = [*x+hsin(wt −kx), hcos(wt −kx)]T , where w = gk.*

Ocean Model – Phase II

- Ocean surface with sum of n trochoid wave trains
- Three sub-models.



Ocean Model – Phase II

- Model hierarchy
  - Average positions
    - Compute inside a grid cell by filtering the trochoids

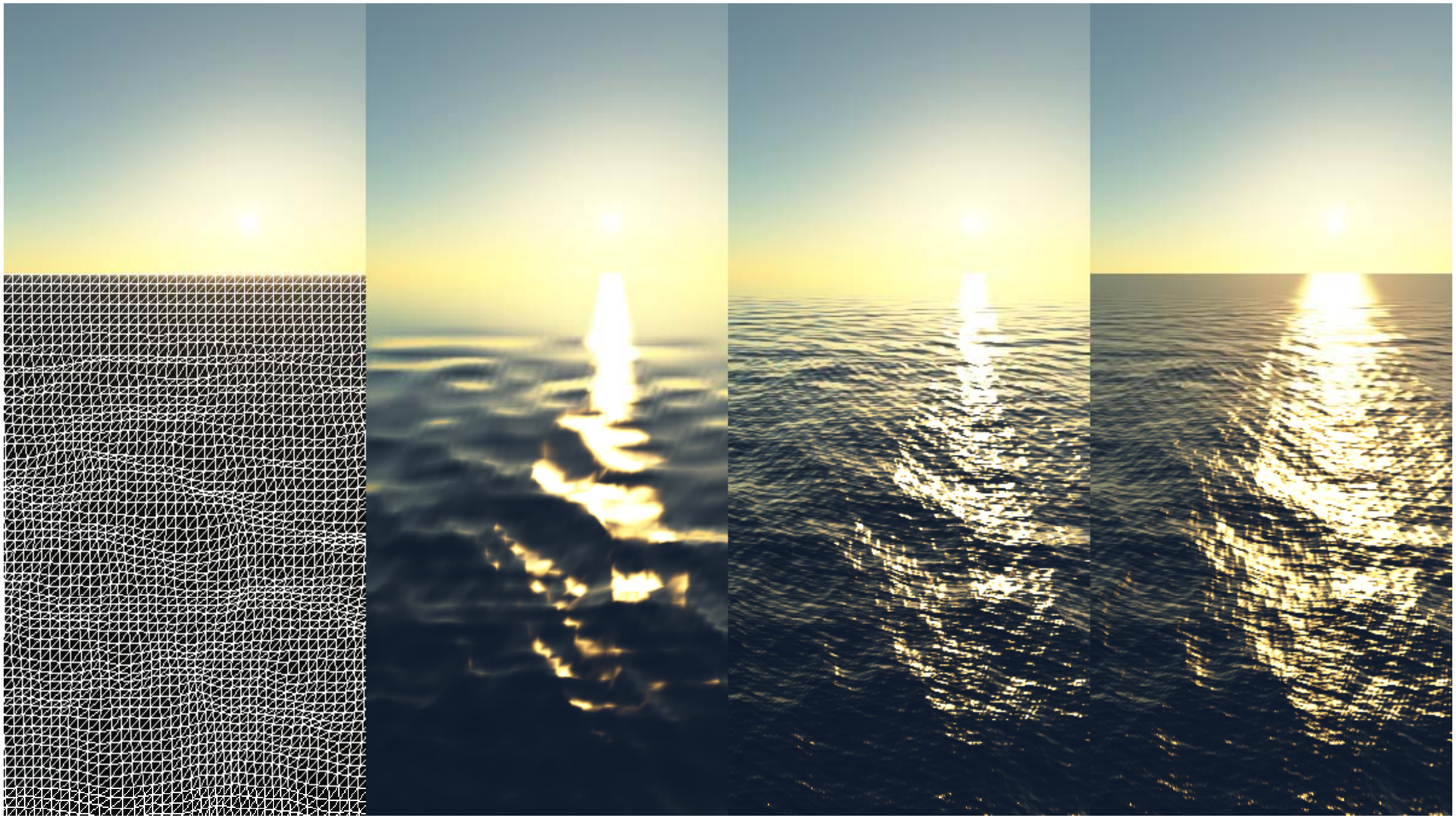
$$\mathbf{p} = \begin{bmatrix} \mathbf{x} \\ 0 \end{bmatrix} + \sum_{1}^{n} w_{p,i} \mathbf{t}_i, \quad \mathbf{t}_i = \begin{bmatrix} \frac{\mathbf{k}_i}{\|\mathbf{k}_i\|} h_i \sin(\omega_i t - \mathbf{k}_i \cdot \mathbf{x}) \\ h_i \cos(\omega_i t - \mathbf{k}_i \cdot \mathbf{x}) \end{bmatrix}$$

  - Average normals
    - Compute inside a pixel

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

  - BRDFs
    - Subpixel surface details with statistical properties

Ocean Model – Result



Ocean BRDF

- A very accurate BRDF model for anisotropic rough surfaces.

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

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

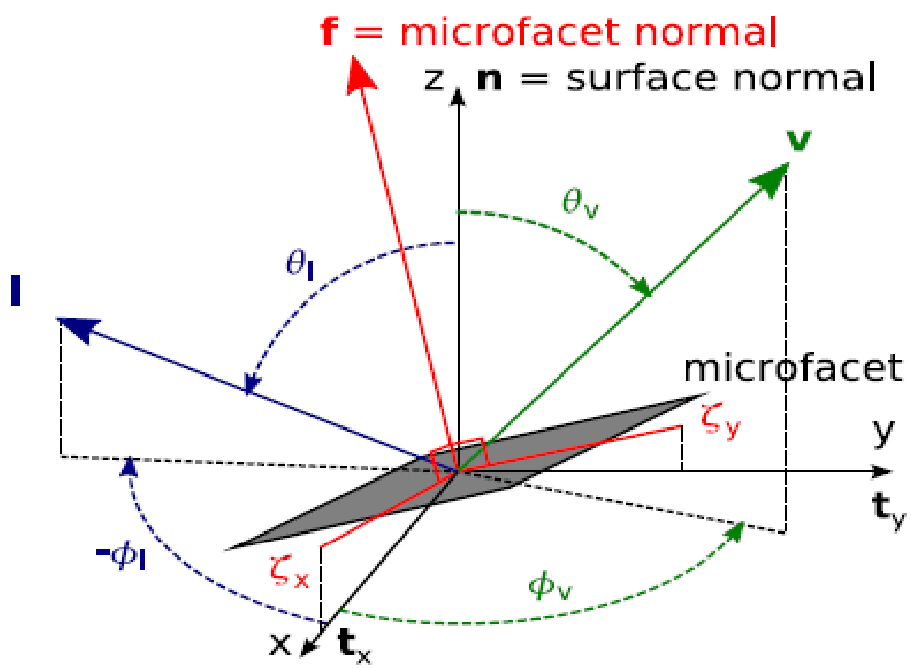
$$p(\boldsymbol{\zeta}) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left(-\frac{1}{2}\left(\frac{\zeta_x^2}{\sigma_x^2}+\frac{\zeta_y^2}{\sigma_y^2}\right)\right)$$

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

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

Ocean BRDF

- BRDF model coordinates
  - v and l are unit vectors towards the viewer and the light. f is the normal of a microfacet whose x and y



Ocean Lighting – Sun Lighting

- Compute the light reflected from the Sun at P by applying the BRDF
- BRDF as constant over the Sum solid angle Ωsun

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

- Self-shadowing can be provided with a shadow map for close views

Ocean Lighting – Sky Lighting

- Light reflected from the sky dome is difficult
- Approximate method for specular to diffuse BRDFs assuming an isotropic or anisotropic Gaussian slope distribution
- Three steps:
  - Approximate environment lighting
  - Average Fresnel reflectance
  - Average sky radiance

Sky Lighting – Approximate environment lighting

- BRDF is proportional to the fraction of micro-facets

$$\mathrm{brdf}(\mathbf{v},\mathbf{l}) = p(\boldsymbol{\zeta}_h)\rho(\mathbf{v},\mathbf{l})$$

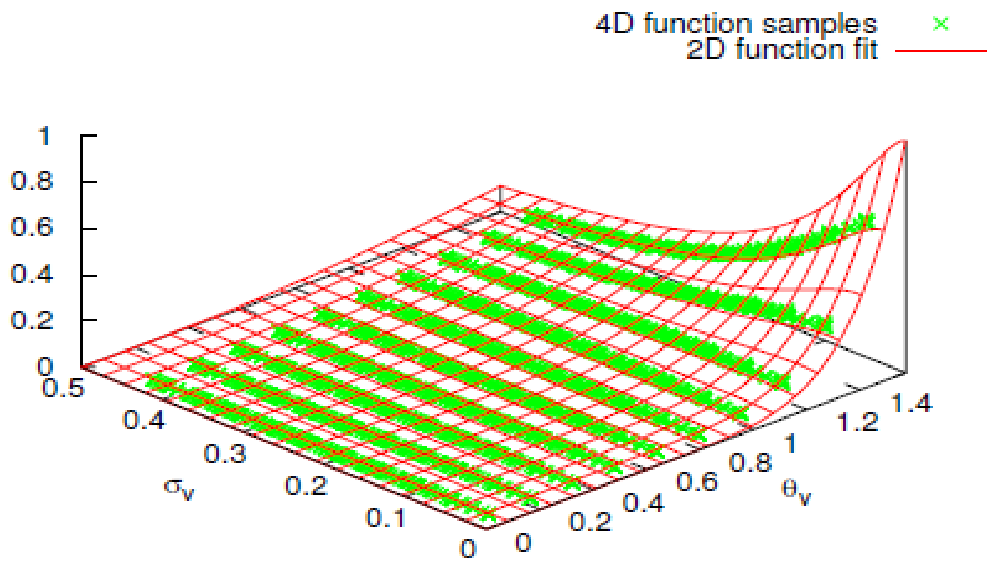
- Approximation is exact when BRDF is purely specular

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

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

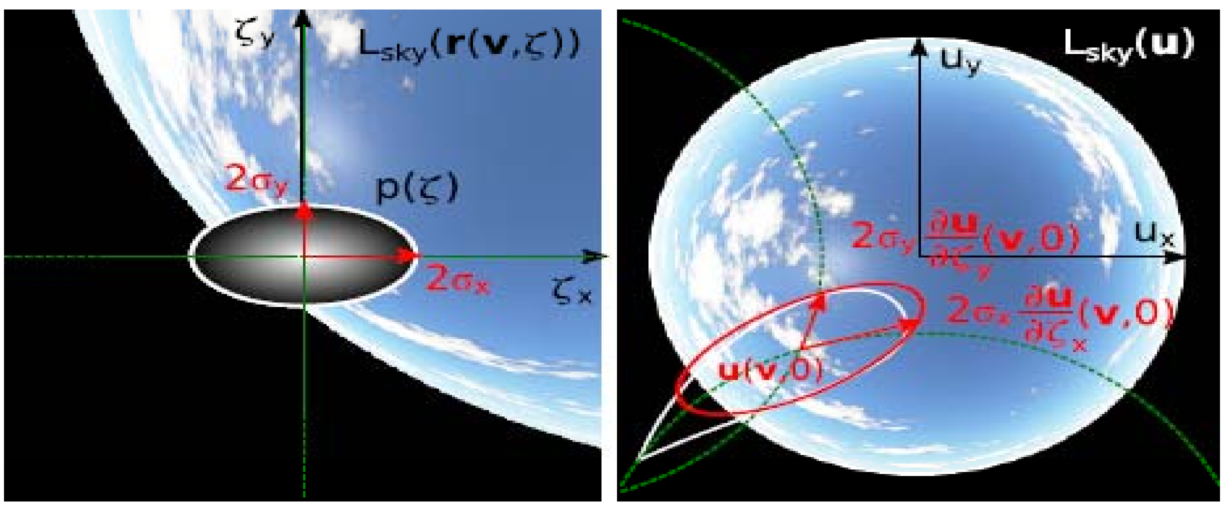
Sky Lighting – Average Fresnel reflectance

- Plot of the reflectance of anisotropic rough surface (green), and filter function (red)



Sky Lighting – Average sky radiance

- Environment map filtering
- The reflected light L is an elliptical Gaussian filter
- Environment map transformed filter



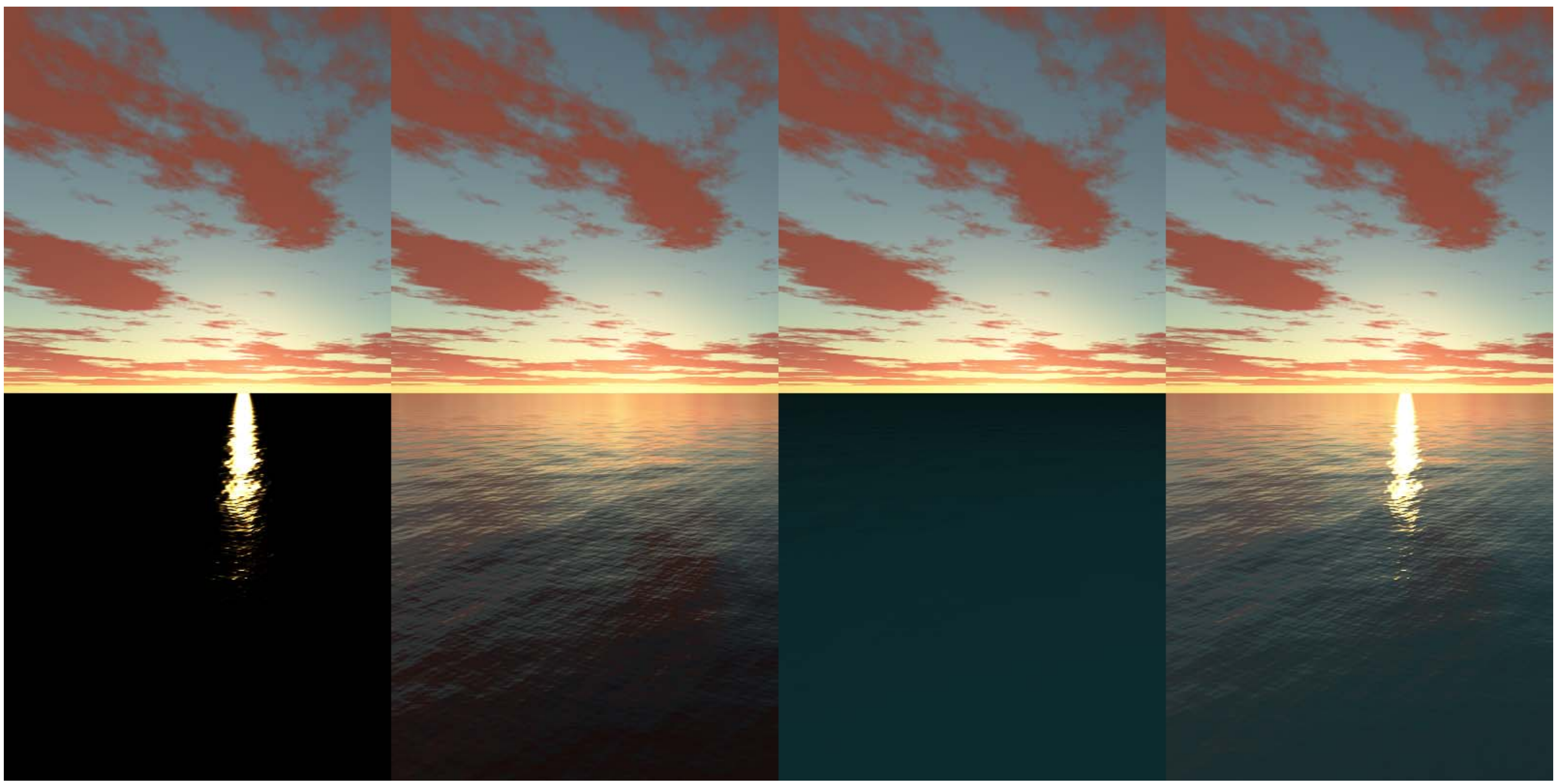
Ocean Lighting – Refracted Lighting

- Light coming from the Sun and Sky also refracted inside the water
- Also refracted again to the viewer
- Radiance Lsea reaching the surface from below is diffuse

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

Ocean Lighting – Result

- Reflected sun light, reflected sky light, light refracted from the water to final result



Summary of Lighting Algorithm

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

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

$\mathbf{f} \leftarrow \mathrm{normalize}([-\zeta_x \;\; -\zeta_y \;\; 1])$ *// 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 $[r_x \;\; r_y]/(1 + r_z)$

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

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

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

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

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

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

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

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

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

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

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

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

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

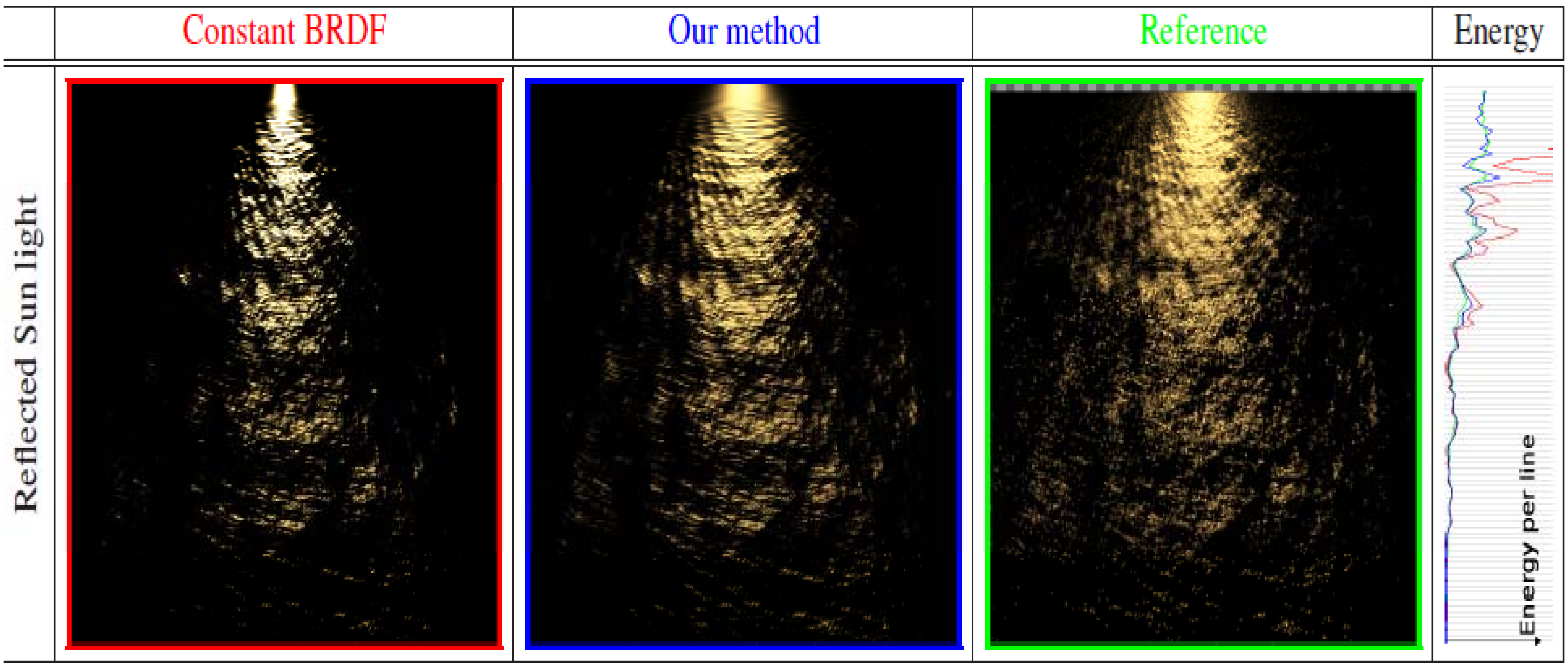
Extensions

- Local waves
  - Support other waves than trochoids
- Local reflections
  - Use reflection map in screen space
- Multiple reflections
  - Environment map approximate sky irradiance
- Planet-scale rendering
  - Render a sphere with Ross BRDF

Implementation

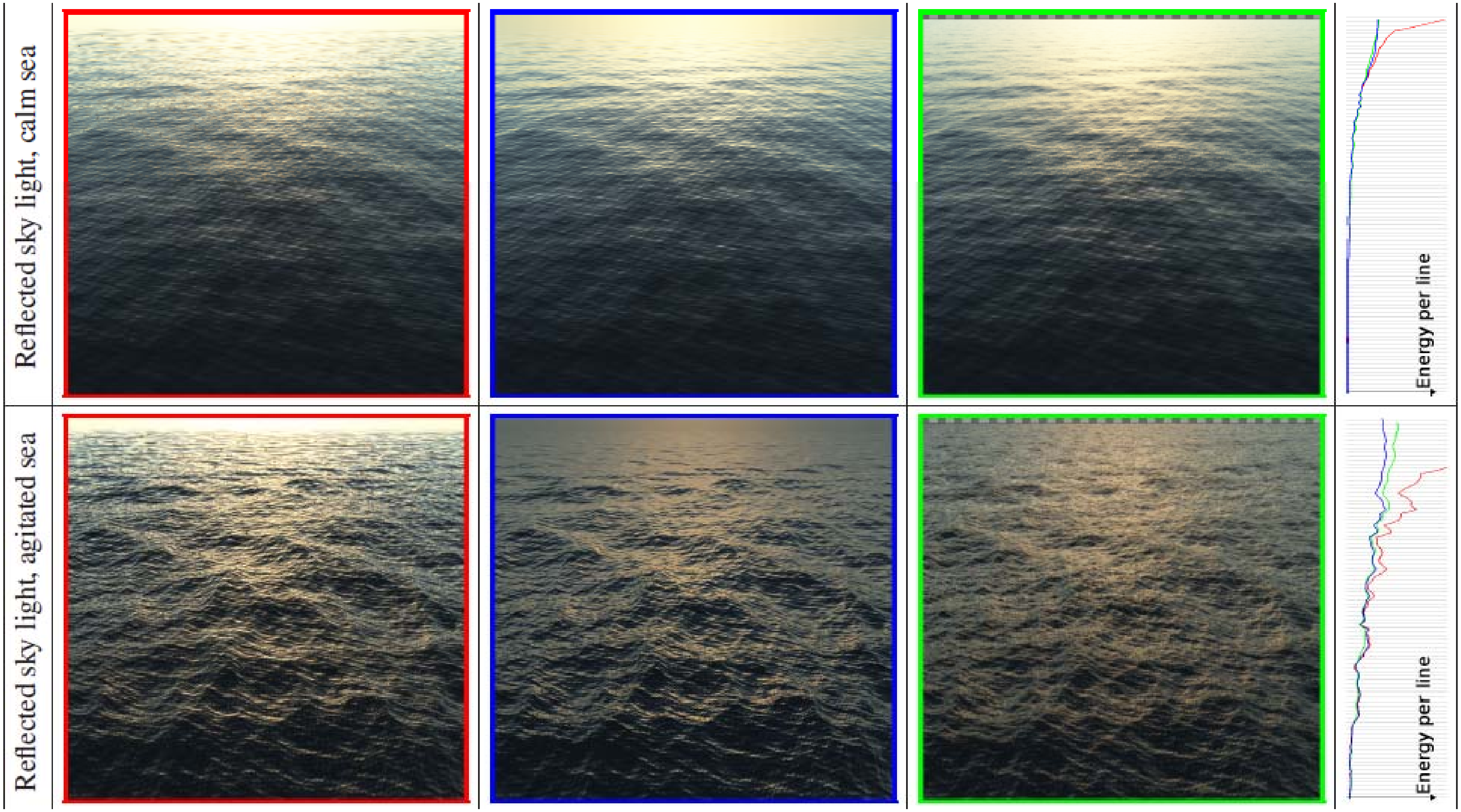
- Vertex shader projects the screen space regular grid
- Fragment shader computes the per pixel normals and the Sun, Sky and refracted light
- Use a geometric progression for the wavelengths

Result



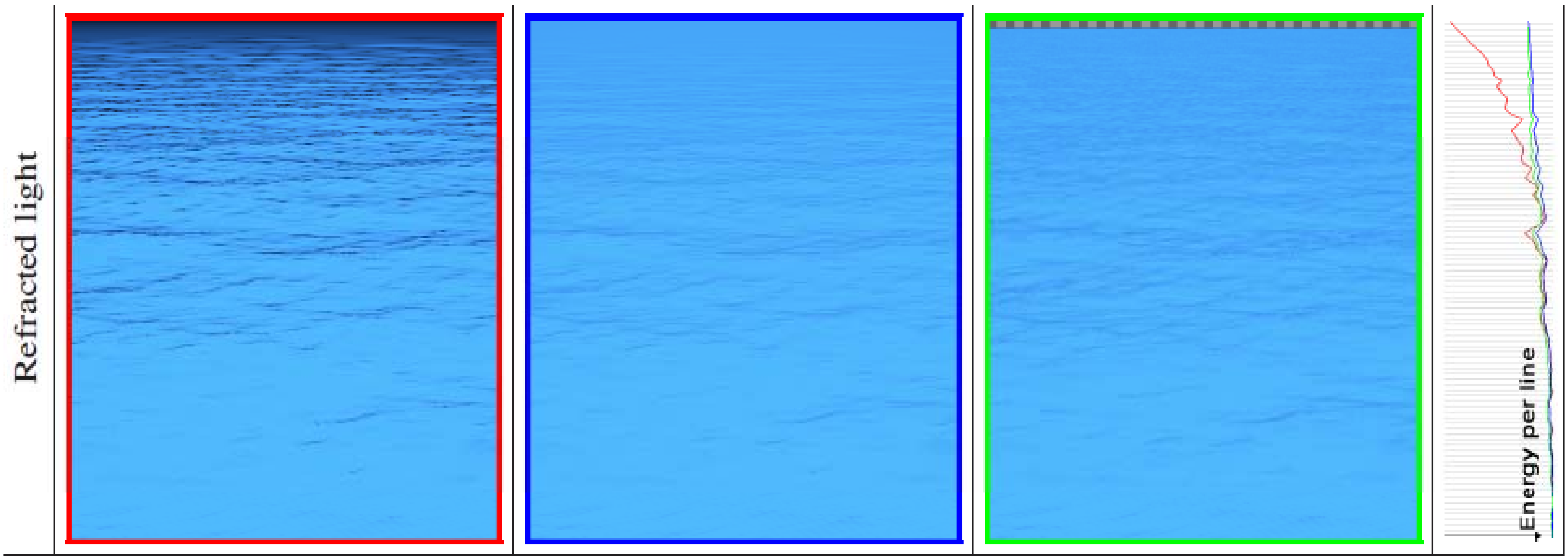
Constant BRDF
Our method
Reference
Energy
Reflected Sun light
Energy per line

Result



Reflected sky light, calm sea
Reflected sky light, agitated sea
Energy per line
Energy per line

Result



Compare to real photo

References

- *An anisotropic phong BRDF model. Ashikhmin M., Shirley P. Journal of Graphics Tool 5(2000)*
- *GPU-based real-time simulation and rendering of unbounded ocean surface. Yang X., Pi X., Zheng L., Li S. In International Conference on Computer Aided Design and Computer Graphics (2005)*
- *Simulating ocean water. Tessendorf J. ACM SIGGRAPH course notes (2001)*