

遮蔽物を考慮した海面のレンダリング

角田 賢亮[†] 岩崎 慶[‡] 高木 佐恵子[‡] 吉本 富士市[‡]

[†]和歌山大学大学院 システム工学研究科 [‡]和歌山大学 システム工学部

概要

CG の分野において、海のレンダリングは重要な研究課題の一つとして様々な手法が提案されてきた。従来の海のレンダリングでは、海面上に遮蔽物がない場合の海を対象としており、洞窟などによって入射光が遮蔽される場合のレンダリング法は提案されていない。本稿では、遮蔽物を考慮した海のレンダリング手法を提案する。水中での遮蔽情報を前計算し、球面調和関数で表現することにより、効率的に水中から視点への輝度を計算する。その結果、遮蔽された海を写実的に表現することができた。

A Rendering Method for Sea Surfaces Taking into Account Occlusion

Kensuke Kakuta[†] Kei Iwasaki[‡] Saeko Takagi[‡] Fujiichi Yoshimoto[‡]

[†]Graduate School of Systems Engineering, Wakayama University

[‡]Department of Systems Engineering, Wakayama University

Abstract

In the field of Computer Graphics, rendering of the sea surfaces is one of the most important research topics, and many methods have been proposed to render the sea surfaces. Previous methods have been proposed to render sea surfaces without occlusions. There exists no method for rendering sea surfaces occluded by objects such as the sea in the cave. In this paper, we present an efficient rendering method of sea surfaces occluded by objects. Our method precomputes the occlusion information and represents the occlusion information with spherical harmonics. This makes it possible to calculate the outgoing radiance from the water surface efficiently. As a result, we got possible to render sea surfaces occluded by objects realistically.

1 はじめに

CG による海の表現に関する研究は、CG の分野において重要な研究課題の一つである。海の CG 表現に関する研究は、波の形状モデリング、水面の色の計算、水中のレンダリングなど幅広い分野で研究が進められてきた。海のレンダリングに関する従来法では、通常海面上に遮蔽物のない海を対象としている。

一方、イタリアのカプリ島に存在する青の洞窟に代表されるように、洞窟などによって海面への入射光が遮られる海が存在する。この場所では、近年、地球温暖化による海面の上昇により、その素晴らしい景観の消滅が懸念されている。そのため、青の洞窟などの貴重な景観をデジタルアーカイブとして残す研究は重要であるといえる。

3次元CGによるデジタルアーカイブは、映像・建物などの人工物を対象とした研究例は多いが、洞

窟などの自然物（自然文化財）を対象とした研究例は少ない。3次元CGで表現すると、光の状態を変化させながら、さまざまな視点から洞窟内を見ることが出来る。したがって、実際の洞窟観光では見ることができないシーンを、仮想的に見ることが出来るので、写真やビデオでは得られないおもしろさを体験できると考えられる。

しかしながら、従来の海のレンダリング法では洞窟などの遮蔽物を考慮したレンダリング手法は提案されていない。そこで本研究では、洞窟などの遮蔽物を考慮した海面のレンダリング法を提案する。本研究では、入射光として太陽光と天空光を考慮する。

本稿の構成は以下のとおりである。2節では関連研究を紹介し、3節では、本手法の概要を述べる。4節、5節では、具体的に水面上・水中の光をどのように扱うかを述べる。6節で適用例を示し、最後に7節でまとめと課題について述べる。

2 関連研究

Fourier ら [1] は海洋表面のモデルを提案した。このモデルは、水分子のパーティクルを楕円軌道で記述し、軌道のパラメータにより現実的な波形を簡単に発生させることができる。また、時間がモデルに組み込まれているためアニメーションが簡単であり、レイトレーシングなど、従来のレンダリング手法が利用できる。Peachey[2] は、傾斜のある砂浜での碎波のレンダリングとアニメーションに適したモデルを提案した。Peachey は、さらに、波の破壊と障害物の衝突で生じた水しぶきをパーティクルシステムを用いてモデル化した。水面を表示するスキャンラインアルゴリズムでは、パーティクルシステムとその他の水面を別に描いて統合している。Ts'o ら [3] は、表面張力による形状のパラメータを用いた B スプラインで水面の複雑な外観を表現している。Ts'o のモデルでは、水中における波の屈折を扱っており、屈折による波の進行方向の変化はスネルの法則で制御している。Mastin ら [4] は水深の深い海における波のモデル化を行った。Watt は、光と水の相互作用に対するバックワードレイトレーシング法 [5] を提案しているが、Watt の方法は水中の複雑な光学性質を考慮していない。Nishita らは太陽光による海中の散乱光の計算法 [6] を提案した。この論文では、大気よりも外側の非常に高い視点から、大気中の散乱を含む水底までの地表のレンダリングを行っている。Premeze らは物理則に従い、水面の色を正確に計算する手法 [7] を提案した。この論文では、数学的手法を用いた穏やかな波のモデリング、泡の表現、光の伝播、Jerlov[8] による海のタイプの分類法が紹介されている。Premeze らは、水中から視点に到達する光を、拡散成分と視線に沿った光の成分に分けて求めている。これにより、水中の複雑な光の散乱を簡略化した。

しかし、これらの手法では、遮蔽物の存在しない海を対象としており、青の洞窟のような遮蔽された海のレンダリングにそのまま適用することは難しい。そこで、本稿では光の遮蔽を考慮した海のレンダリング法を提案する。

3 概要

水面に入射する光には、図 1 に示すとおり、太陽光と天空光がある。前者は、一方向からの直射日光を指し、後者は大気中で散乱した、あらゆる方向から地上に到達する光を指す。遮蔽物の存在により、入射する

光はさえぎられ、水中・水面上での輝度は非一様となる。本手法では遮蔽物による入射光の遮蔽情報を前計算し、球面調和関数で表現する。これにより、水中の輝度分布を効率的に計算することができる。

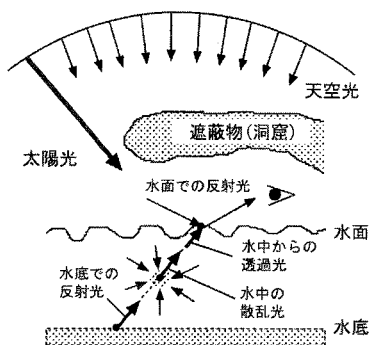


図 1 水面の輝度計算の概念図

水面から視点に到達する光 L_v は水面での反射光成分 L_R と水中からの透過光成分 L_T からなる。

$$L_v = RL_R + TL_T \quad (1)$$

ここで R , T は水面上の点における反射率と透過率であり、フレネルの法則により計算される。

L_R は太陽光による反射光 L_R^{sun} と天空光による反射光 L_R^{sky} に分けられ、さらに、 L_T は散乱光と水底での反射光に分けられる。 L_T においても入射光を太陽光と天空光に分けて計算する。

以降の節において、各要素の計算方法について述べ、最終的に水面から視点に到達する光 L_o を計算する。前提として、反射光の計算では、水面は凹凸を考慮して計算し、透過光の計算では、水面は水平と仮定する。これは、透過光は反射光に比べ、波の形状による光の影響が比較的少ないからである。本手法では、水中の散乱光について 1 次散乱を考慮する。

4 反射光 L_R の計算

水面上の点 x から視点に向かう反射光 L_R は以下のように計算する。まず、太陽光による反射光 L_R^{sun} は以下の式で表される。

$$L_R^{sun}(x, \omega) = L^{sun}(\omega_{sun})V(x, \omega_{ref})\delta(\omega_{sun}, \omega_{ref}) \quad (2)$$

ここで、 L^{sun} は太陽光の放射輝度、 ω_{sun} は太陽光の入射方向、 ω_{ref} は視線方向 ω の反射方向とし、 $V(x, \omega_{ref})$ は水面上の点 x における ω_{ref} 方向の可視関数で、 x から ω_{ref} 方向のレイが遮蔽物と交差する場合は 0、それ以外は 1 を返す関数とする。 $\delta(\omega_{sun}, \omega_{ref})$

は ω_{ref} と ω_{sun} が同じ方向の場合のみ 1, それ以外は 0 を返すデルタ関数とする (図 2 参照).

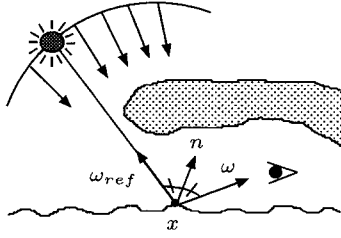


図 2 反射光の計算方法

一方, 天空光による反射光 L_R^{sky} は,

$$L_R^{sky} = L^{sky}(\omega_{ref})V(x, \omega_{ref}) \quad (3)$$

で計算される. ここで, $L^{sky}(\omega_{ref})$ は ω_{ref} 方向の天空光の輝度, $V(x, \omega_{ref})$ は前述の可視関数とする. $L^{sky}(\omega_{ref})$ は環境マップで表現する.

太陽光の反射光 L_R^{sun} と天空光の反射光 L_R^{sky} は, GPU のフラグメントプログラムにより計算する. スクリーンの各ピクセルに対応する水面上の点 x において反射方向 ω_{ref} を計算し, 可視関数 $V(x, \omega_{ref})$ は, x から ω_{ref} 方向にレイを飛ばし遮蔽物 (洞窟) との交差判定を行って計算する.

5 透過光 L_T の計算

水面上の点 x に向かう水中の透過光の輝度 $L_T(x, \omega')$ は以下の式で求められる.

$$L_T(x, \omega') = L_{bottom}(P, \omega') \exp(-\sigma_t L) + \int L_s(p, \omega') \exp(-\sigma_t l) dp \quad (4)$$

ここで, ω' は視線方向 ω の屈折方向, P は屈折方向と水底との交点, p は屈折した視線上の点とする (図 3 参照).

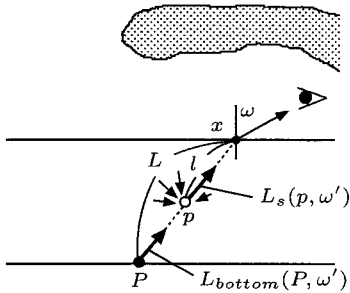


図 3 透過光 $L_T(x, \omega')$ の計算

第 1 項が水底での点 P における反射光 $L_{bottom}(P, \omega')$ が距離 L だけ光が減衰して視点に向かう光の輝度, 第 2 項が水中で屈折した視線 (方向 ω') 上の点 p における ω' 方向への散乱光 $L_s(p, \omega')$ を, 視線に沿って積分した輝度である.

5.1 水底における反射光

水底の点 P における反射光 L_{bottom} は, 太陽光成分 L_{bottom}^{sun} と天空光成分 L_{bottom}^{sky} からなる.

$$L_{bottom}(P, \omega') = L_{bottom}^{sun}(P, \omega') + L_{bottom}^{sky}(P, \omega') \quad (5)$$

まず, 太陽光成分 L_{bottom}^{sun} は以下の式で計算する.

$$L_{bottom}^{sun}(P, \omega'_{sun}) = TL^{sun}(\omega_{sun}) \times f_r \exp(-\sigma_t L')V(P, \omega'_{sun})(n_P \cdot \omega'_{sun}) \quad (6)$$

ここで, T は水面の点 P' における太陽光の透過率 (図 4 参照), $L^{sun}(\omega_{sun})$ は ω_{sun} 方向の太陽光の輝度, ω'_{sun} は ω_{sun} の屈折方向, σ_t は水中における光の減衰係数, L' は点 P' から ω'_{sun} 方向へ屈折した光が点 P に到達したときの距離, $V(P, \omega'_{sun})$ は水底の点 P における可視関数, f_r は水底の反射率, n_P は水底の法線とする.

水底の点 P における可視関数 $V(P, \omega'_{sun})$ は以下のようにして計算される. 点 P から ω'_{sun} 方向のレイを飛ばし, 水面との交点 P' を計算する. P' において ω_{sun} 方向のレイを飛ばし, 遮蔽物との交差判定によって $V(P, \omega'_{sun})$ を計算する (図 4 参照).

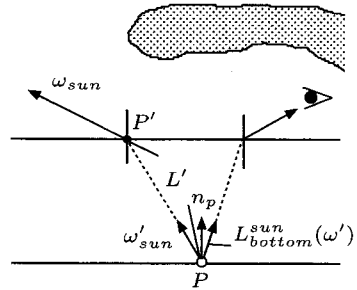


図 4 太陽光による水底での反射光 $L_{bottom}^{sun}(P, \omega')$

天空光成分 L_{bottom}^{sky} は以下の式で計算する.

$$L_{bottom}^{sky}(P, \omega') = \int_{\Omega} L^{sky}(\omega_{sky})E(P, \omega'_{sky})d\omega'_{sky} \quad (7)$$

$$E(P, \omega'_{sky}) = V(P, \omega'_{sky})T \exp(-\sigma_t l')f_r(n_P \cdot \omega'_{sky}) \quad (8)$$

ここで, $L^{sky}(\omega_{sky})$ は ω_{sky} 方向の天空光の輝度, ω'_{sky} は ω_{sky} の屈折方向, $V(P, \omega'_{sky})$ は水底の点 P における可視関数, l' は水面から ω'_{sky} 方向へ屈折した光が

点 P に到達したときの距離, f_r は水底の反射率, n_p は水底の法線とする (図 5 参照)。

本研究では, 視点を移動できるようにするため, 任意の位置で L_{bottom}^{sky} を計算する必要がある。 L_{bottom}^{sky} を計算するためには, 任意の位置において様々な方向にレイを飛ばし, 式 (7) を積分計算する。 視点が移動するたびに L_{bottom}^{sky} の積分計算をするのは計算コストが非常に高い。 この計算を高速化するために, 水底にサンプル点を設定し, 各サンプル点において可視関数 V をあらかじめ計算しておく。 近隣のサンプル点から P における可視関数を補間により求める。 さらに, 入射光 L と $V(p, \omega'_{sky})$ を球面調和関数で表現することにより積分計算を効率よく処理する。

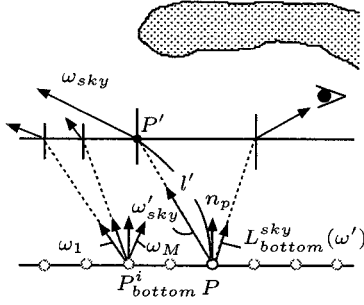


図 5 天空光による水底での反射光 $L_{bottom}^{sky}(P, \omega')$

図 5 のように, 水底上に N 個のサンプル点 $P_{bottom}^i (1 \leq i \leq N)$ を格子状に設定する。 各 P_{bottom}^i において, 上半球上に M 個のサンプルレイ $\omega_j (1 \leq j \leq M)$ を生成する。 生成したレイ ω_j を飛ばし, 水面において屈折したレイと遮蔽物との交点計算を行う。 交差した場合には $V(P_{bottom}^i, \omega_j) = 0$ とし, 交差しない場合は $V(P_{bottom}^i, \omega_j) = 1$ として $E(P_{bottom}^i, \omega'_{sky})$ を計算する。 $E(P_{bottom}^i, \omega)$ を球面調和関数展開し, 4 次の球面調和係数を用いて E を表現する。

$$E(P_{bottom}^i, \omega) = \sum_{k=1}^{16} E_k(P_{bottom}^i) y_k(\omega) \quad (9)$$

ここで, $y_k(\omega)$ は k 番目の球面調和関数とし, 係数 $E_k(P_{bottom}^i)$ を各 (P_{bottom}^i) において保存する。 $L_{bottom}^{sky}(P, \omega')$ は, P の近隣のサンプル点 P_{bottom}^i における係数から補間した係数 $E_k(P)$ と L の球面調和係数の内積によって効率的に計算することができる。

5.2 水中の散乱光

式 (4) の第 2 項を解析的に積分することは難しいため, 屈折した視線上に離散的な点を設定し, 数値積分

により求める。 $L_s(p, \omega')$ を太陽光成分 L_s^{sun} と天空光成分 L_s^{sky} に分けて別々に計算する。

5.2.1 散乱光の太陽光成分

散乱光 $L_s(p, \omega')$ の太陽光成分 L_s^{sun} は以下の式で計算される。

$$L_s^{sun}(p, \omega') = TL^{sun}(\omega_{sun}) \exp(-\sigma_t l_p) \times \rho(\omega'_{sun}, \omega') V(p, \omega'_{sun}) \quad (10)$$

$\rho(\omega'_{sun}, \omega')$ は ω'_{sun} 方向から ω' へ散乱する割合を表す位相関数, $V(p, \omega'_{sun})$ は水中の点 p における可視関数, l_p は水面の点 P' から点 p までの距離とする (図 6 参照)。 ここで, $V(p, \omega'_{sun})$ は視点と太陽光の位置が変化しない限り, 一度計算するだけでよい。

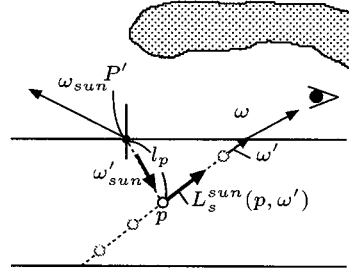


図 6 散乱光の太陽光成分 $L_s^{sun}(p, \omega')$

5.2.2 散乱光の天空光成分

$L_s(p, \omega')$ の天空光成分 L_s^{sky} は以下の式で計算される。

$$L_s^{sky}(p, \omega') = \int_{\Omega} L^{sky}(\omega_{sky}) E(p, \omega'_{sky}) \times \rho(\omega'_{sky}, \omega') d\omega'_{sky} \quad (11)$$

$$E(p, \omega'_{sky}) = V(p, \omega'_{sky}) T \exp(-\sigma_t l_p) \quad (12)$$

ここで, $\rho(\omega'_{sky}, \omega')$ は位相関数, $V(p, \omega'_{sky})$ は水中の点 p における可視関数, l_p は水中の点 P' から点 p までの距離とする (図 7 参照)。

式 (11) の計算は計算コストが高いため, 前計算により高速に計算する。 式 (12) の $E(p, \omega'_{sky})$ は減衰率 σ_t が変わらない限り変化しないため, 水中の離散的な点において前計算することができる。

図 7 のように, K 個のサンプル点 $p^i (1 \leq i \leq K)$ を水中に設定し, 各 p^i について, 上半球上の M 個のサンプルレイ $\omega_j (1 \leq j \leq M)$ を飛ばし, 水面において屈折したレイと遮蔽物との交差判定を行い, $V(p, \omega'_{sky})$ を計算し, $E(p^i, \omega)$ を球面調和関数展開する。

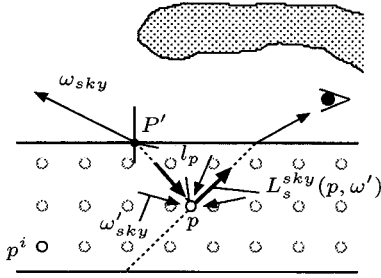


図7 散乱光の天空光成分 $L_s^{sky}(p, \omega')$

$$E(p^i, \omega'_{sky}) = \sum_{k=1}^{16} E_k(p^i) y_k(\omega'_{sky}) \quad (13)$$

また、位相関数 $\rho(\omega'_{sky}, \omega')$ についてもあらかじめ球面調和関数展開しておく。

$$\rho(\omega'_{sky}, \omega') = \sum_{k=1}^{16} \rho_k(\omega') y_k(\omega'_{sky}) \quad (14)$$

最終的に、 $L_s^{sky}(\omega_{sky})$ 、 $E(p, \omega'_{sky})$ 及び $\rho(\omega'_{sky}, \omega')$ の球面調和係数のトリプルプロダクト [9] によって、式 (11) の積分計算を効率よく処理することができる。

6 結果

本手法により生成した画像例を図8~10に示す。ここで、水深は5m、前計算の水中のサンプル点数 K は $32 \times 32 \times 8$ 、水底のサンプル点数 N は 32×32 、散乱光の数値積分のための視線の分割数は32、水中における光の減衰係数 σ_t は [6] を参考に設定した。画像サイズは 640×480 pixel、水面の格子点数は 256×256 とした。計算環境は、CPUがPentium4 3.4GHz、メモリが1GB RAM、GPUにGeForce 7800 GTXを搭載したPCを用いた。この条件でレンダリングを行った結果、1フレームの描画に1.484秒要した。

天空光の前計算（遮蔽情報の取得・球面調和関数展開）や散乱光成分の計算、太陽光の散乱光成分の計算はCPU上で行い、水面上での反射光のレイトレーシングや水底での反射光の計算、その他のレンダリング処理はGPUで行った。表1に天空光の前計算時間を示す。サンプルレイの数 M は $32 \times 32 \times 3$ とし、水深の分割数を変化させている。

表1 天空光の散乱光成分の前計算処理に要する時間

サンプル点数	前計算時間 [sec]
$32 \times 32 \times 2$	2.266
$32 \times 32 \times 4$	4.547
$32 \times 32 \times 8$	11.42
$32 \times 32 \times 16$	53.31

7 まとめ

本稿では、遮蔽物を考慮した海のレンダリング手法を提案した。遮蔽物を考慮した水面での反射光、水中での散乱光、水底での反射光を計算する手法を述べた。本手法では、入射光として太陽光及び天空光を考慮することにより、写実的な海の画像を生成することができる。水中における遮蔽情報を球面調和関数で表現することにより、水中の散乱光及び水底での反射光を効率的に計算することができた。

今後の課題として、CPUで計算した部分のGPU化による高速化が挙げられる。

参考文献

- [1] A. Fournier and W. T. Reeves, "A simple model of ocean waves", Computer Graphics (SIGGRAPH '86 Proceedings), Vol.20, No.4, pp.75-84, 1986.
- [2] D.R. Peachey, "Modeling waves and surf", Computer Graphics (SIGGRAPH '86 Proceedings), Vol.20, No.4, pp.65-74, 1986.
- [3] P.Y. Ts'o and B.A. Barsky, "Modeling and rendering waves: wave-tracing using beta-splines and reflective and refractive texture mapping", ACM Transactions on Graphics, Vol.6, No.3, pp.191-214, 1987.
- [4] G.A. Mastin, P.A. Watterberg and J.F. Mareda, "Fourier synthesis of ocean scenes", IEEE Computer Graphics and Applications, Vol.7, No.3, pp.16-23, 1987.
- [5] M. Watt, "Light, water interaction using backward beam tracing", Computer Graphics, Vol.24, No.4, pp.377-385, 1990.
- [6] T. Nishita, T. Shirai, K. Tadamura, and E. Nakamae, "Display of the earth taking into

account atmospheric scattering”, Proc. SIG-GRAPH’93, pp.175-182, 1993.

[7] S. Premoze and M. Ashikhmin, “Rendering natural waters”, Computer Graphics Forum, Vol.20, No.4, pp.189-199, 2001.

[8] N.G. Jerlov, “Marine optics”, Elsevier, Amsterdam, 1976.

[9] R. Ng, R. Ramamoorthi, and P. Hanrahan, “Triple product wavelet integrals for all-frequency relighting”, ACM Transactions on Graphics, Vol.23, No.3, pp.477-487, 2004.



図8 洞窟内の海面のレンダリング画像

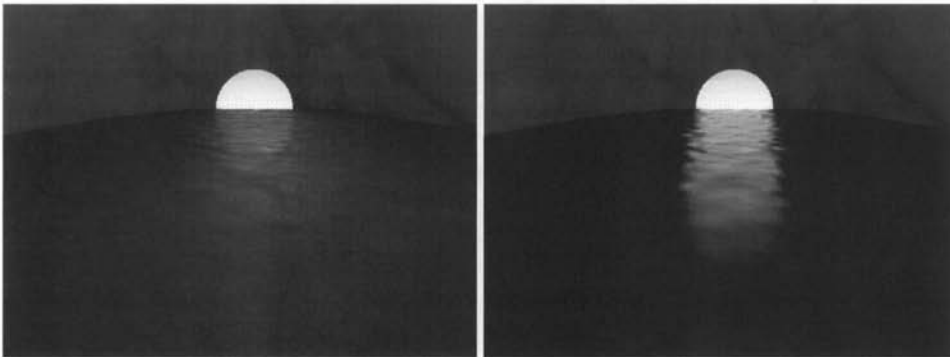


図9 透過光 L_T (左: 天空光成分 L_T^{sky} 右: 太陽光成分 L_T^{sun})

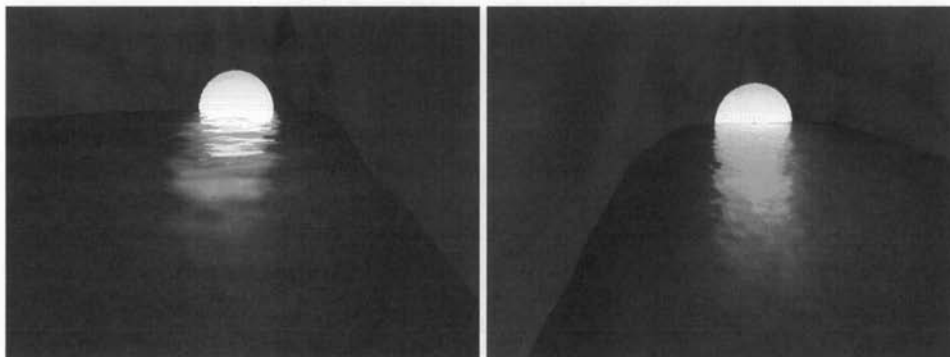


図10 視点を変更した例 (左: 海面付近の視点から見た例 右: 視点の高度を高くした例)