

# グラフィックスハードウェアを用いた水の色および水中の光跡の高速表示法

岩崎 慶<sup>†</sup> 土橋 宜典<sup>††</sup> 西田 友是<sup>†</sup>

<sup>†</sup> 東京大学 <sup>††</sup> 北海道大学

## 1 はじめに

近年、グラフィックスハードウェアの低価格化と高性能化が進んでおり、これを利用した写実的な画像の表示法の研究が行われている [1]。また、自然物の写実的表現の中で、水面や水中の景観は魅力的な CG 画像として いくつかの研究がなされてきた [2][3]。そこで、本稿ではグラフィックスハードウェアを用いて、水の色および水中における光跡を高速にレンダリングする手法を提案する。

## 2 提案手法の概要

水に関連した自然物 (海や湖) あるいはプール等の画像を CG を用いて生成する際、視点の位置によって考慮すべき点や現象が異なる。

### 2.1 視点が水面より上にある場合

水面の色を計算するうえで考慮すべき光の要素は、水中での散乱光、水底で反射した透過光、水面での反射光の 3 つである。水面に入射する光は、太陽直達光と天空光がある。水面での反射光は太陽直達光と天空光を考慮し、散乱光、透過光は太陽直達光のみを考慮するものとする。また、散乱光と透過光については、水面は平面と仮定して計算する。散乱光に関しては 2 次散乱光まで考慮する手法が提案されている [4]。

水中で 2 次散乱する光を計算するためには、視線上にサンプル点を設定してその点に入射してくる 1 次散乱光を全方位について積分する必要がある。提案法では、水面と平行な仮想平面を考え、視線と仮想面の交点にサンプル点を配置する。そして、仮想面上の各点に入射する 1 次散乱光を画像として記憶する。仮想面上の各点から視点に到達する散乱光は、その画像に体積散乱関数 (これも画像として記憶する) を乗じたものを加算することによって計算することができる。つまり、従来法は点単位であるのに対し、提案法は面単位で計算が行われる。また、グラフィックスハードウェアはポリゴン、すなわち面単位での処理を基本としている。そのため、提案法ではグラフィックスハー

ドウェアを利用して高速に計算することができる。

### 2.2 視点が水中の場合

水中での光学的効果には、水分子による散乱光を考慮した水の色に加え、光跡があり、筆者の一人がスキャンラインアルゴリズムを利用した表示法を開発した [2]。本稿ではこの従来法を改良し、グラフィックスハードウェアを利用した高速な光跡の表示法を提案する。

水中における光跡は、波によって屈折した光によってできる。これは水面の波がレンズのように働くため、屈折光が集光することによって生じる現象である。一様でない密度の透過物体を可視化するには数値積分が必要だが、水中での密度は一定なので、輝度は距離の解析的関数 (指数関数) で計算できる。水面に入射する光は太陽直達光と天空光があるが、光跡については太陽直達光のみからなるものとし、天空光は環境光とみなす。水面の表示では散乱光は 2 次散乱までを考慮したが、視点が水中の場合 1 次散乱のみを考慮する。また、光跡に焦点を置くため水中に物体はないものとし、物体での反射光や物体による影は考慮しないものとする。

## 3 水面の色の計算

### 3.1 従来法

水中の散乱光については、筆者の一人が 2 次散乱まで考慮した方法を提案している [4]。しかしながら、この方法では多くの計算時間を必要とする点が問題となっている。水中の 2 次散乱光成分を計算するためには、視線上にサンプル点を設定し、そのサンプル点に入射する 1 次散乱光  $I_s^{(1)}$  に体積散乱関数  $\beta(\alpha)$  を乗じたものを全方位について積分する必要がある。したがって、水中の点  $P_k$  (深さ  $z_k$ ) から視点に到達する 2 次散乱光  $I_s^{(2)}$  は以下のような式で表される (図 1)。

$$I_s^{(2)}(z_k) = \int_0^{2\pi} \int_0^\pi \beta(\alpha(\theta, \phi)) G(z_k) I_s^{(1)}(\theta, \phi, z_k) \sin \theta d\theta d\phi \quad (1)$$

ここで、 $G(z)$  は深さ  $z$  の点で散乱された光が水面に到達するまでの減衰率であり、 $\exp(-c(\lambda)z \sec \theta_{ij})$  で計算される。また  $I_s^{(1)}$  は  $(\theta, \phi)$  方向から深さ  $z$  の点  $P_k$  に入射する 1 次散乱光を表す。式 (1) は解析的に計算することができないので、 $n_w$  の方向に離散化して計算

A Method of Fast Rendering of Water and Shafts of Light using Graphics Hardware

<sup>†</sup> Kei Iwasaki, Tomoyuki Nishita

<sup>††</sup> Yoshinori Dobashi

University of Tokyo (<sup>†</sup>)

Hokkaido University (<sup>††</sup>)

する。すなわち、離散化した方向  $(\theta_i, \phi_i) (i = 1, \dots, n_w)$  から  $P_k$  (深さ  $z_k$ ) に入射する 1 次散乱光を求め、以下のような式で総和を計算する。

$$G(z_k) \sum_{i=1}^{n_w} \beta(\alpha(\theta_i, \phi_i)) I_s^{(1)}(\theta_i, \phi_i, z_k) \sin \theta_i \Delta \theta \Delta \phi \quad (2)$$

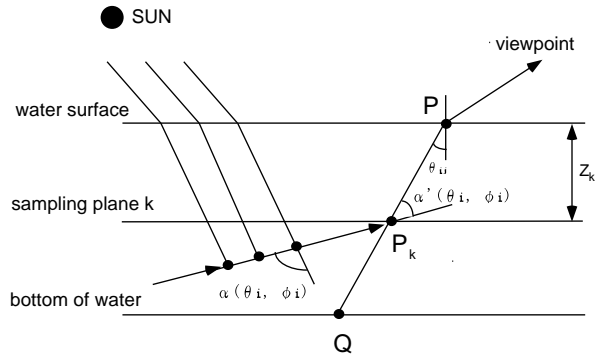


図 1: *sampling plane* を使った水面のレンダリング法

### 3.2 提案法

図 1 に示すように、水面と平行な仮想平面 (*sampling plane* と呼ぶ) を考え、視線と *sampling plane* の交点にサンプル点を配置する。1 次散乱光  $I^{(1)}(\theta_i, \phi_i, z_k)$  は深さと方向にしか依存しないため、同じ *sampling plane* 上のサンプル点ならば同じ値をとる。サンプリング点を 1 画素とみなすと、この *sampling plane* における 1 次散乱光は同一輝度の画像 ( $I_{image}^{(1)}$ ) とみなすことができる。また、体積散乱関数  $\beta(\alpha(\theta_i, \phi_i))$ 、減衰率  $G(z_k)$  も同様に画像とみなすことができる (ただし、この 2 つの画像はサンプル点ごとに違う値をとるため同一色ではない)。*sampling plane k* のすべてのサンプリング点から視点までの 2 次散乱光は、まず  $n_w$  枚の  $I_{image}^{(1)}$  と体積散乱関数を画像とみなした  $\beta_{image}$  をそれぞれ掛け合わせる。次に  $n_w$  枚の掛け合わされた画像を足し合わせた結果得られる画像に減衰率を画像とみなした  $G_{image}$  を掛け合わせることによって計算される。画像同士の掛け算は OpenGL のブレンディング関数を使用し、画像の足し算は *Accumulation Buffer* を使用する。このように式 (2) の計算を、グラフィックスハードウェアを用いることができる。

## 4 水中における水の色の計算

視点が水中にある場合の水の色の計算法は [2] を基にしている。視点  $P_v$  に到達する光の経路は以下の 3 つに分類される (図 2)。

1. 視線と水面が交差し、水面の法線と視線とのなす角が臨界角 (48.6 度) より小さい場合 (図 2 の  $P_1$ )、入射光が視点に到達する。

2. 視線と水面が交差し、水面の法線と視線とのなす角が臨界角より大きい場合 (図 2 の  $P_2$ )、水面で全反射した光が視点に到達する。
3. 視線が水面と交差しない場合。通常水底等と交差する。水の減衰係数  $c(\lambda)$  は一般に高く、 $\exp(-c(\lambda)L_\infty) < \epsilon$  を満たす距離  $L_\infty$  よりも遠いところで交差した光は無視するものとする。

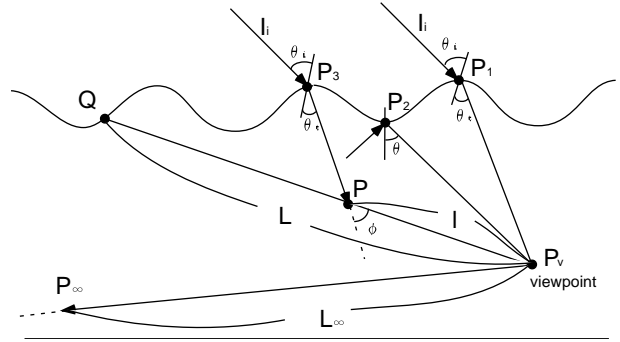


図 2: 視点に到達する光の経路

光路 1. の場合、水面上の点  $Q$  から視点  $P_v$  に入射する光  $I_v(\lambda)$  は以下の式で計算される。

$$I_v(\lambda) = I_Q(\lambda) \exp(-c(\lambda)L) + \int_0^L I_P(\lambda) \exp(-c(\lambda)l) dl \quad (3)$$

ここで、 $c(\lambda)$  は水の減衰係数、 $L$  は  $P_v Q$  間の距離、 $l$  は  $PP_v$  間の距離、 $I_P$  は点  $P$  での散乱光を表す。 $I_Q$  は、屈折角  $\theta_t$  が臨界角よりも小さい場合、以下の式で与えられる。

$$I_Q = T(\theta_i, \theta_t) (I_i(\lambda) \delta(\theta_i, \theta_t) + I_{sky}(\theta_i)) \quad (4)$$

$I_i(\lambda)$  は入射光の輝度、 $\delta(\theta_i, \theta_t)$  はデルタ関数 (屈折方向と視線方向が一致したとき 1、それ以外は 0)、 $I_{sky}(\theta_i)$  は天空光の輝度とする。光路 2. の場合、全反射した光を環境光とみなし、これに  $P_2 P_v$  間の減衰率を乗じたものと  $P_2 P_v$  間で散乱した光を積分したものと和が、 $P_2$  から  $P_v$  に入射する光の輝度となる。光路 3. の場合、 $L_\infty$  よりも遠いところからの光は無視できるため、点  $P_\infty (P_v P_\infty = L_\infty)$  から  $P_v$  まで散乱光を積分したものが  $P_v$  に入射する光の輝度となる。

## 5 水中の光跡の表示法

### 5.1 従来法

水中の光跡の輝度計算は、前節の散乱光の積分項 (式 (3) の第 2 項) を計算することによって求められる。水面は、格子分割し、細分化した三角形メッシュと考える。各格子点における法線ベクトルと太陽光の入射ベクトルによって決まる屈折ベクトルを sweep するこ

とによってできる volume を *Illumination volume* とし、走査面との交差部分の三角形を *caustic triangle* とする (図 3)。

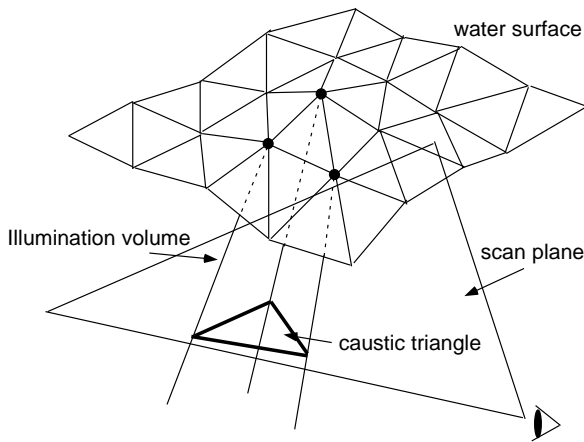


図 3: *Illumination volume* と *caustic triangle*

ルギーは、対応する水面の三角形と *caustic triangle* との面積比に反比例する。*caustic triangle* の輝度は  $I_p d_i \exp(-c(\lambda)l_i)$  で計算される。 $I_p$  は *caustic triangle* の、単位長さあたりの平均輝度、 $d_i$  は視線と *caustic triangle* との交差する長さ、 $l_i$  は *caustic triangle* と視点との距離とする。各走査線について *caustic triangle* の輝度を計算し *Accumulation Buffer* に貯えることによって光跡が表示される。しかしながらこの方法は、各走査線ごとに可視領域内の全 *caustic triangle* の輝度を計算するため、多くの時間が必要となる問題点がある。

## 5.2 提案法

提案法は従来法 [2] を基にしている。提案法でも *Illumination volume* を作成する。*Illumination volume* をいくつかの volume に分割し、これを *Sub-volume* と呼ぶ (図 4)。従来法では *Illumination volume* を走査面ごとに切って散乱光 (つまり *caustic triangle* の輝度) を計算していたのに対し、提案法では *Sub-volume* ごとに輝度を計算し、描画する。*Sub-volume* は、まず 3 つの四面体に分割され、次に各四面体をスクリーンへ三角形ポリゴンの集合として描画される。ポリゴンの描画についてはグラフィックスハードウェアでの高速化が図れる。

### 5.2.1 *Sub-volume* の輝度計算

*Sub-volume* の各頂点の輝度を以下のように求める。これは、*Sub-volume* の各頂点と対応する水面上の点における輝度を予め計算しておき、これに水面から *Sub-volume* の頂点までの減衰率を掛けたものと、三角形 (図 5 では  $P_1P_2P_3$ ) の面積比を掛けて計算する。

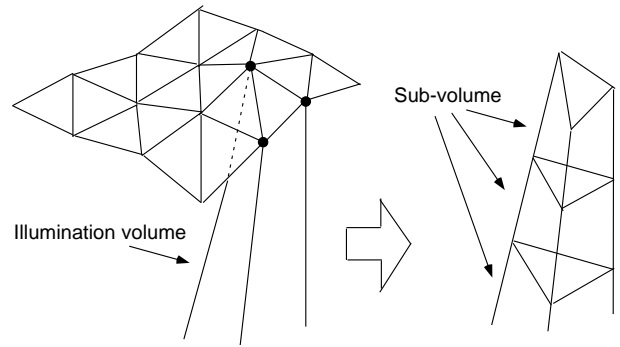


図 4: *Illumination volume* と *Sub-volume*

次に *Sub-volume* を 3 つの四面体に分割する (図 5)。各四面体をスクリーンに投影し、次に述べる 2 つ (Case1 と Case2) に場合分けして描画される。

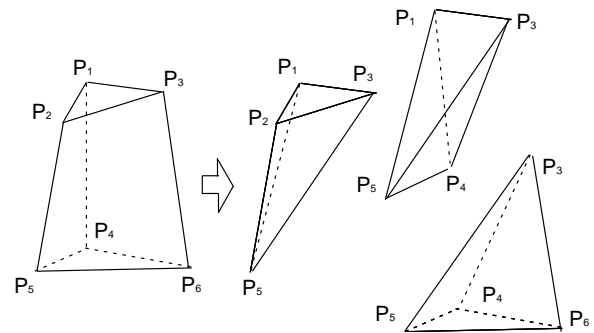


図 5: *Sub-volume* の 3 つの四面体への分割

### 5.2.2 Case 1

視点と  $P_2$  を通る直線と  $P_1P_3P_5$  からなる平面との交点  $P$  を求める。Case1 は、図 6 のように交点  $P$  が三角形  $P_1P_3P_5$  の内部にある場合である。 $P$  における輝度は  $P_1, P_3, P_5$  の輝度から補間して求める。 $P_2P$  間の散乱光は、 $P_2P$  間で輝度が線形に変化するとして視線が  $P_2P$  を通過した時の累積輝度  $C$  を計算によって求まる。そして  $P_2$  の輝度を  $C$ 、その他の頂点での輝度を 0 として三角形  $P_2P_1P_5, P_2P_3P_5$ , および  $P_2P_1P_3$  をスムーズシェーディング機能を用いて描画する。

### 5.2.3 Case2

Case2 は、図 7 のように視点と  $P_2$  を通る直線と  $P_1P_3P_5$  からなる平面との交点が三角形  $P_1P_3P_5$  の外部にある場合である。視点と  $P_1P_3$  からできる平面と線分  $P_2P_5$  の交点  $P_a$  を求める。視点と  $P_2P_5$  からできる平面と線分  $P_1P_3$  の交点  $P_b$  を求める。 $P_2$  および  $P_5$  の輝度から  $P_a$  の輝度を、 $P_1$  および  $P_3$  の輝度から  $P_b$  の輝度をそれぞれ補間して求める。 $P_aP_b$  間は輝度が線形に変化するとして視線が  $P_aP_b$  を通過したときの

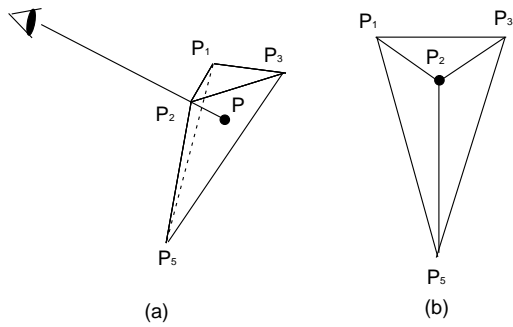


図 6: Case1:(a) 視点と四面体との位置関係 (b) 投影図

散乱光の累積輝度  $C$  を計算する。そして  $P_2$  の輝度を  $C$ 、その他の頂点での輝度を 0 として 4 つの三角形  $P_2P_1P_a$ ,  $P_2P_3P_a$ ,  $P_1P_aP_5$ , および  $P_aP_3P_5$  をスムーズシェーディング機能を用いて描画する。

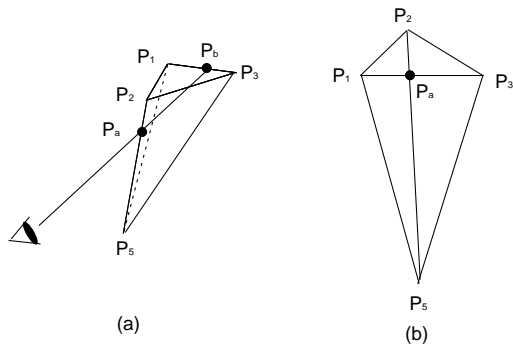


図 7: Case2:(a) 視点と四面体との位置関係 (b) 投影図

## 6 計算結果

図 8 および図 9 にそれぞれ水面および水中の光跡の表示結果を示す。波形の生成には [5] の方法を利用した。計算環境は Dell Precision420(PentiumIII 1GHz)、グラフィックボードは Intense3D Wildcat4110 を使用している。実行時間は水中の光跡では 28 秒要した。三角形メッシュの総数は 131072( $256 \times 256 \times 2$ ) とした。水面の表示では、*sampling plane* の数が 10、離散化した方向の数  $n_w$  が 128、サンプル点の数が 4096 のとき、0.8 秒要した。同じ計算をソフトウェアで行った場合、1.4 秒要した。

## 7 まとめと今後の予定

グラフィックスハードウェアを利用することにより、ソフトウェアで計算する場合よりも高速にレンダリングすることができた。またこの方法ならば、グラフィックスハードウェアの進歩にともなって更なる高速化が期待される。

今後の予定は、光跡だけでなく、物体上に生じる

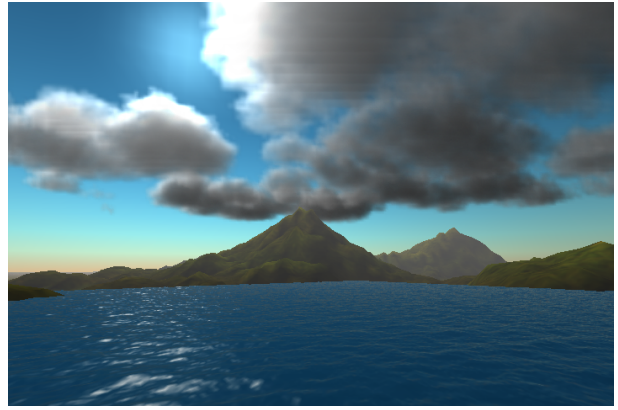


図 8: 水面の表示



図 9: 水中の光跡

*caustics* や水中の物体によってできる影をグラフィックスハードウェアを使用して高速にレンダリングすることである。

## 参考文献

- [1] Y.Dobashi, T. Yamamoto, T.Nishita, "Interactive Rendering Method for Displaying Shafts of Light," Proc. of Pacific Graphics 2000, 2000-10, pp. 31-37.
- [2] T. Nishita, E. Nakamae, "Method of Displaying Optical Effects within Water using Accumulation-Buffer," Proc. of SIGGRAPH'94, 1994-7, pp.373-380.
- [3] T.Nishita, T.Shirai, K.Tadamura, E.Nakamae, "Display of The Earth Taking into account Atmospheric Scattering," Proc. of SIGGRAPH'93, 1993-8, pp.175-182.
- [4] 岩崎 西田: 多重散乱を考慮した海面の色の表示法, 情報処理学会第 61 回全国大会, 2000
- [5] J. Tessendorf, "Simulating Ocean Water", SIGGRAPH'99 Course Note. Simulating Natural Phenomenon, pp. 1-18.