

GPUを用いた空のフルスペクトルレンダリング

Full-spectrum rendering of the sky using GPU

高橋憂雅¹, 土橋宜典¹Yuga TAKAHASHI¹, Yoshinori DOBASHI¹¹ 北海道大学¹ Hokkaido University

1 はじめに

近年、映画やゲームといった分野でCGにより写実的な映像の生成を行うために様々な研究が行われている。リアルな映像を生成するためには、色の表現が重要である。現在、多くのレンダリングシステムでは計算過程から結果の出力に至るまでを光の三原色であるRGBを使用して行う。RGBレンダリングは一見して問題の無いように見えるが、波長依存性の高い光学現象を正しくシミュレーションすることができない。そのため、物理的に正しいレンダリングを行うためには、光のスペクトルを考慮してレンダリングを行う必要がある。しかし、スペクトルレンダリングを行うためには、RGBレンダリングに比べて計算を行う波長の数が多くなってしまったため、計算コストが増えてしまうという問題点がある。

本稿では、上記の問題に対してスペクトル分布を多項式により表現することで、より少ないサンプルからフルスペクトルの画像を効率的に生成する手法を提案する。

2 先行研究

空の表示に関する手法は数多く提案されている。土橋らは一定間隔でサンプリングされた太陽高度ごとに輝度分布を事前に計算することで、高速に表示する手法を提案している [1]。しかし、この手法ではフルスペクトルでの計算は行われていない。近年では、空の輝度分布を解析的なモデルにより表現する手法が提案されており、フルスペクトルでのレンダリングも可能である [2]。しかし、この手法では、大規模なデータベースが必要となる。提案手法は、GPUを用いた並列計算と多項式近似を利用して高速にフルスペクトルの空の輝度を求める。

3 提案手法

3.1 概要

提案手法の概要を図1に示す。空の色の計算は大気による光の散乱光を視線に沿って積分することで求めら

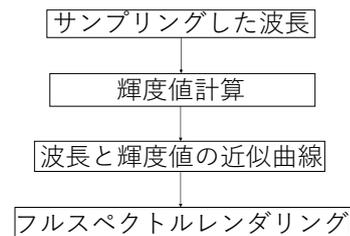


図1 提案手法の概要

れる。可視光範囲の全波長にわたって輝度計算を行うことは計算コストが高い。我々の事前実験により、空の輝度と波長の関係は滑らかで3次多項式により精度よく近似できることがわかっている。そこで、提案法では、指定された4つの波長について輝度計算を行い、その結果から波長と輝度値の関係を表す3次多項式を求める。ただし、輝度計算においては、一次散乱のみを考慮する。

3.2 空の色の計算

大気は、空気分子とエアロゾルの両方から構成されている。そのため、それぞれの粒子による散乱を考慮した輝度値の計算を行う必要がある。輝度値の計算は次の式により行う。[1]

$$L(\lambda) = \frac{I_s(\lambda)F(\alpha, \lambda)}{\lambda^4} L_s(\lambda) \quad (1)$$

$$L_s(\lambda) = \int_s \rho(s) \exp(-\tau(s, \lambda)) ds \quad (2)$$

$$\tau(s, \lambda) = \frac{4\pi k}{\lambda^4} \int_t \exp(-\frac{h(t)}{H_0}) ds \quad (3)$$

ここで、 λ は波長、 $I_s(\lambda)$ は大気外からの太陽光の強度、 $F(\alpha, \lambda)$ は位相関数である。 s は視点から視線上の一点までの距離、 ρ は密度比、 τ は光路長を表す。

3.3 近似曲線を用いたレンダリング

指定された4波長について、式(1)(2)(3)を用いて輝度計算を行い、その結果から波長と輝度値の関係を表す

3次多項式を求める。そして、その3次多項式を用いて任意の波長での輝度値を求めることで、計算時間を短縮する。以下にその概要をまとめる。

まず、指定された4波長について、各ピクセルの輝度計算をGPUにより行う。計算結果はテクスチャとして保存され、すべての画素について計算した後、CPUへ転送する。次に、各画素について、4つの波長とその輝度値から三次関数の係数を求める。これは、波長と輝度値の関係を表す連立一次方程式を解くことで求めることができる。

以上により、任意の波長の空の輝度を求めることができるが、ディスプレイに表示する際には、フルスペクトルからRGBに変換する必要がある。本稿では、以下に示すように、XYZ表色系に変換した後、RGB値を求めた。

$$X = \frac{1}{k} \int_{\lambda} S(\lambda) I_{\bar{x}}(\lambda) d\lambda \quad (4)$$

$$Y = \frac{1}{k} \int_{\lambda} S(\lambda) I_{\bar{y}}(\lambda) d\lambda \quad (5)$$

$$Z = \frac{1}{k} \int_{\lambda} S(\lambda) I_{\bar{z}}(\lambda) d\lambda \quad (6)$$

$$k = \int_{\lambda} I_{\bar{y}}(\lambda) d\lambda \quad (7)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} \approx \begin{bmatrix} 3.24 & -1.54 & -0.50 \\ -0.97 & 1.88 & 0.042 \\ 0.056 & -0.20 & 1.06 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (8)$$

$S(\lambda)$ は近似曲線で得たスペクトル、 $I(\lambda)$ はXYZ表色系のそれぞれの等色関数を表す。

4 結果と考察

提案法の計算精度を確認するための実験を行った。波長を400nmから30nm間隔で12波長をサンプリングして輝度計算を行った場合と提案法を用いた場合とで誤差計算と計算時間の比較を行った。使用した計算機は、Intel Core i5-8250U 1.60GHz(CPU) および Intel UHD Graphics 620(GPU) を搭載したノートPCである。

提案法においては、430nm, 520nm, 610nm, 700nmなる4つの波長での輝度を計算して3次多項式を求めた。仰角0°と仰角90°における空の画像を図2に示す。

輝度計算を行う際の太陽高度を仰角0°, 30°, 45°, 60°, 90°として実験を行った。誤差評価は、各画素の輝

表1 提案法の誤差

仰角	0°	30°	45°	60°	90°
誤差	0.020	0.032	0.031	0.030	0.028

度値の平均二乗誤差 (RMSE) により行った。表1に輝度の値の誤差を示す。太陽光の輝度はすべての波長で1.0としている。

計算時間についても比較を行った。各画素について、12波長の輝度値をすべて求めるためにかかる時間を計測した。提案法は186.38秒、12波長をサンプリングして輝度計算する場合は408.05秒であった。提案法により2倍以上の高速化を実現できることがわかった。

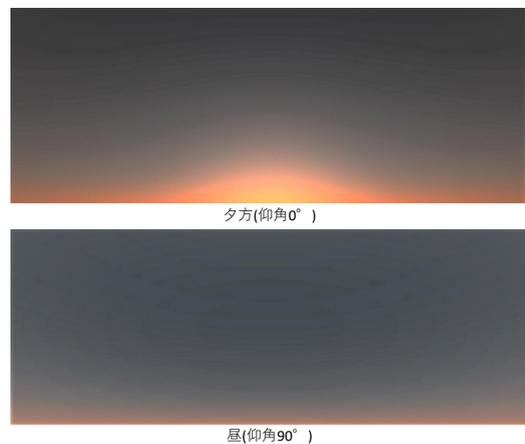


図2 計算例

5 まとめ

本研究では、より少ないサンプルから近似多項式を求めて、任意の波長に対応する輝度の値を求めることにより空のフルスペクトルレンダリングを効率的に行った。計算時間においては、半分ほどに短くすることができた。しかし、近似多項式を求めるためにサンプル波長によって誤差が大きくなってしまいう問題点が挙げられる。

参考文献

- [1] Y. Dobashi et al., "Fast Display Method of Sky Colour Using Basis Functions", The Journal of Visualization and Computer Animation, VOL. 8: 115-127 (1997)
- [2] A. Wilkie et al., "A fitted radiance and attenuation model for realistic atmospheres," ACM Trans. Graph., vol. 40, no. 4 (2021).