

## 雲のボリュームレンダリングの編集システム

岩崎 航<sup>†1</sup> 土橋 宜典<sup>†1</sup> 山本 強<sup>†1</sup>

本研究では、ボリュームデータで表現された雲の表示について着目する。一般に、ボリュームデータはボリュームレンダリングを行うことで可視化することができるが、その結果は多くのパラメータに依存し、ユーザが所望する結果を直感的に生成することは難しい。本稿では、雲のレンダリングにおけるパラメータを直感的に指定することで、所望のレンダリング結果が容易に得られるシステムを提案する。提案システムでは、ユーザは、レンダリングパラメータを直接指定するのではなく、より直感的な操作により間接的に指定する。本システムを用いることで、ユーザは雲の輝度計算式を意識することなく、直感的に所望のボリュームレンダリング結果を生成することが可能である。

## An Editing System for Volume Rendering of Clouds

WATARU IWASAKI,<sup>†1</sup> YOSHINORI DOBASHI<sup>†1</sup>  
and TSUYOSHI YAMAMOTO<sup>†1</sup>

In this research, we focus on the display of clouds represented by volume data. Generally, volume data can be visualized by a volume rendering technique. However, there are many parameters that have to be specified by the user and it is often difficult to create the desired image by adjusting those parameters. In this paper, we propose a system that allows the user to specify the parameters intuitively and to generate the desired results easily. In our system, the user does not directly specify the rendering parameters, but implicitly specify them by intuitive operations. By using our system, the user can intuitively create the desired result without understanding the complex equations for the intensity of clouds.

<sup>†1</sup> 北海道大学大学院情報科学研究科  
Graduate School of Information Science and Technology, Hokkaido University

### 1. はじめに

コンピュータグラフィックス技術の発展に伴い、これまで様々な物体表現手法が提案されてきた。本研究では、そのような物体表現手法の一つであるボリュームデータで表現された雲に着目する。一般的に、雲のボリュームデータはボリュームレンダリングを行うことで可視化することができる。しかし、その結果は多くのレンダリングパラメータに依存し、ユーザの所望する結果を得るためには細かなパラメータ調整作業が必要となることが多い。雲の場合、直射光成分、環境光成分、アルベド、減衰率などのパラメータによって結果が大きく左右される。これらのパラメータ調整作業は直感的ではないばかりか、各パラメータがボリュームレンダリングの結果にどのような影響を及ぼすかといった、パラメータに対する専門的な知識を必要とする。そこで、本研究では、パラメータに対する専門的な知識を持たないユーザであっても直感的に雲のボリュームレンダリングを編集することができるシステムの開発を目指す。提案法では、ユーザはパラメータを直接的に指定するのではなく、直感的な編集操作により間接的に指定する。これにより、所望の結果を容易に得ることが可能である。以降、本稿の構成は、2節で関連研究について簡単な紹介を行い、3節で提案手法について述べる。その後、4節で提案手法を用いた実験結果とその考察について述べる。最後に5節でまとめとする。

### 2. 関連研究

レンダリングの編集システムに関して、これまで多くの研究が行われてきた。文献1)では、物体間の映り込みをインタラクティブに編集することができるシステムが提案されている。この手法では、ユーザは直感的に完全鏡面反射方向を変更することができる。完全鏡面反射方向は、物体間の映り込みを決定するパラメータであり、通常、反射の法則に従って定義されるものである。この手法では、反射の法則に関わらず、ユーザが自由に完全鏡面反射方向を決定できるため、非現実的なレンダリング結果を生成することが可能である。文献2)では、シーン中の影や陰影をインタラクティブに編集することができるシステムが提案されている。この手法では、ユーザは直感的かつ局所的に各物体のシェーディング要素を変更することができる。この手法も、文献1)の手法と同様に、非現実的なレンダリング結果を生成することが可能である。文献3)では、環境照明による陰影をインタラクティブに編集することができるシステムが提案されている。この手法では、ユーザは編集したい箇所とそうでない箇所をそれぞれ直感的に指定することで、局所的に陰影を編集することができ

る．この手法は，ユーザの編集に応じて環境照明を変更しており，物理計算に基づいたレンダリング結果を生成する．提案法では，文献 3) の手法のように，ユーザの編集に応じて雲のレンダリングパラメータを間接的に決定することで，物理計算に基づいたレンダリング結果を生成する．

### 3. 提案手法

提案法では，雲の輝度計算式における各パラメータの値をユーザの編集操作に応じて変更することでボリュームレンダリングの編集を行う．まず，本稿で用いている雲の輝度計算式を以下に示す（図 1 参照）．

$$I(p, \lambda) = \int_0^{T(p)} \rho(t) g(t) (I_s(t, \lambda) + I_m(t, \lambda)) dt \quad (1)$$

$$g(t) = \exp(-\kappa\tau(t)) \quad (2)$$

$$\tau(t) = \int_0^t \rho(l) dl \quad (3)$$

ここで， $\lambda$  は波長を表し，RGB によりサンプルする．また， $p$  はピクセル， $t$  はピクセル  $p$  を通る視線上の点と視点との距離， $T(p)$  は雲の厚み， $\rho$  は密度， $I_s(t, \lambda)$  は視線上の一点に到達する直射光成分， $I_m(t, \lambda)$  は環境光成分， $\kappa$  は減衰率を表す．これらのレンダリングパラメータを適切に設定することで，ユーザの所望するボリュームレンダリング結果を生成することが可能である．しかしながら，各パラメータがどのようにボリュームレンダリングに影響するかを直感的に理解することは困難である．例えば，直射光成分を調整することで雲を所望の色に変更したい場合，直射光の色をその所望する色に設定しても，雲の色は所望する色となるとは限らない．また，陰影の付き方を所望のものへと変更したい場合，直射光方向の値を変更し結果を見て再び調整する，といった作業を繰り返さなければならない．環境光成分にいたっては，その設定値によって雲のレンダリング結果がどのように変化するかを予想することは極めて難しい．また，一般に，環境光成分はすべてのボクセルに対して同じ値が格納されるため，単純な値の調整作業だけでは局所的に環境光成分を編集することは不可能である．これらの問題を解決するために，提案法では，三つの編集モードを用いて各パラメータの値を間接的に変更する．すなわち，提案システムは，直射光の色を設定するモード，環境光成分を設定するモード，直射光方向を変更するモードなる三つのモードを有する．ユーザが各モードにおいて編集操作を行うと，それに合わせて上式の中の対応するパラメータが変更される．直射光の色を設定するモードと直射光方向を設定するモードでは

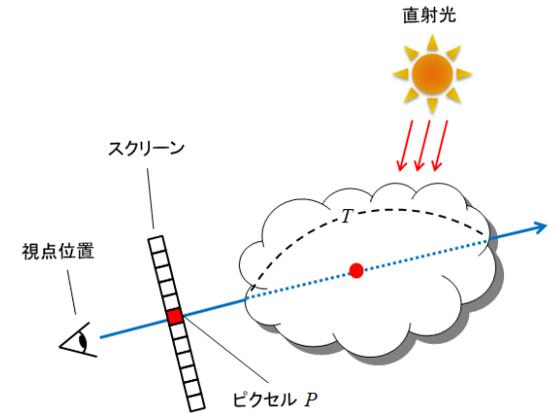


図 1 雲の輝度  
Fig. 1 Intensity of clouds

$I_s(t, \lambda)$ ，環境光成分を設定するモードでは  $I_m(t, \lambda)$  がそれぞれ変更される．以下，各モードについて詳しく説明する．

#### 3.1 直射光の色を設定するモード

このモードでは，ユーザの指定した色情報に合わせて直射光の色を変更する．まず，ユーザは所望の色を選択して，スクリーン上に直接色塗りを行う．そして，ユーザの指定した色情報とレンダリング後の雲の色が最も近くなるように，最適化によって直射光の色を決定する．いま，スクリーン上の  $k$  個のピクセルに対してユーザが色塗りを行ったとすると，直射光の色を求める式は以下ようになる．

$$\min_{I_l(\lambda)} \sum_k (I_{user}(\lambda) - I_{comp}(k, \lambda))^2 \quad (4)$$

ここで， $I_l(\lambda)$  は直射光の色， $I_{user}(\lambda)$  はユーザが指定した色， $I_{comp}(k, \lambda)$  は式 (1) により計算した雲の色である．光源は太陽光のみであるため，直射光の色はすべてのボクセルにおいて同じ値となる．また，雲の明るさは太陽光に比例するため， $\overline{I_{comp}}(k, \lambda)$  を太陽光の輝度が 1 のときの雲の輝度とすると，式 (4) は以下のように書くことができる．

$$\min_{I_l(\lambda)} \sum_k (I_{user}(\lambda) - I_l(\lambda) \overline{I_{comp}}(k, \lambda))^2 \quad (5)$$

上式を  $I_l(\lambda)$  について解くと、式(6)が得られる。これにより、求める直射光の色を算出することが可能である。

$$I_l(\lambda) = I_{user}(\lambda) \frac{\sum_k \overline{I_{comp}}(k, \lambda)}{\sum_k \overline{I_{comp}}^2(k, \lambda)} \quad (6)$$

### 3.2 環境光成分を設定するモード

このモードでは、ユーザが指定した色情報をもとに環境光成分を付加する。ユーザは所望のボクセルに対して所望の環境光成分を設定する。ボクセルへの色塗り方法については、後述するレイを用いた方式を採用する。環境光成分の設定後、文献4)の手法を用いてユーザが設定した環境光成分を他のボクセルへと伝播させる。文献4)の手法を用いることで、ユーザは局所的にボクセルに色塗りを行うだけでなく、効率的に環境光成分を所望の色に設定することができる。前述したように、提案法では、各ボクセルごとに異なる値の環境光成分を格納する。そのため、部分的に異なる環境光成分を付加することが可能であり、従来よりも多様な結果を生成することができる。以下、編集の流れを説明する。

まず、ユーザは所望の色を選択して、雲を表すボクセルへの環境光成分の設定を行う。これは、以下に述べるレイを用いた方式によって行う(図2参照)。図2に示すように、ユーザは、所望の色のストロークを画面上でペイントする。そして、ユーザによりペイントが行われた各画素と視線を結ぶレイを生成し、そのレイと交差する全てのボクセルのうち、雲の密度が0より大きいボクセルにペイントされた色を環境光成分として割り付ける。これにより、ユーザは3.1節と同様に二次元的にスクリーンに対して色塗りを行うだけで、三次元物体である雲に対して効率的に色塗りを行うことが可能となる。

色塗りが行われると、文献4)の手法を用いて色情報を雲全体に伝播させる。提案法では、文献4)の手法をGPUにより実装し、さらに連立一次方程式の高速解法であるマルチグリッド法<sup>5)</sup>を用いて高速化を図っている。本来、文献4)の手法は、グレースケール画像を対象にしてユーザが色塗りを行い、その色情報をグレースケールの輝度分布に従って伝播させるものである。提案法では、これを応用し、環境光成分の伝播に用いる。具体的には、上述の方法によりユーザが指定した環境光成分を、直射光成分の値に従って伝播させる。すなわち、直射光成分が類似した値のボクセルでは、環境光成分も同じであると仮定して、環境光成分を伝播させる。伝播させた環境光成分の大きさは係数により調整できる。この調整作業は、ユーザによりインタラクティブに行われる。調整が終わると、伝播させた環境光成分と各ボクセルの直射光成分を足し合わせ、式(1)により雲の輝度を計算する。

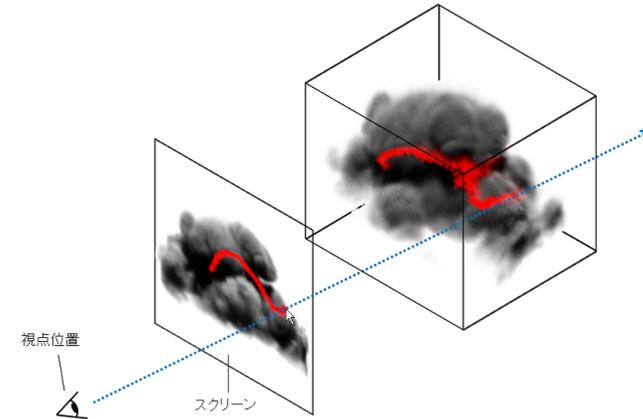


図2 環境光成分のペイント方法  
Fig. 2 A method for painting ambient colors

### 3.3 直射光方向を設定するモード

このモードでは、ユーザのマウスによるドラッグ操作に合わせて直射光の方向を変更することで、陰影を移動させる(図3参照)。ユーザがドラッグ操作を開始したピクセルを  $P'$ 、現在のマウスの位置を表すピクセルを  $Q'$  とする。まず、視点位置から  $P'$ 、 $Q'$  に向かってレイを生成し、それぞれ雲が存在するすべてのボクセルとの交点のうち最も視点位置に近いボクセル ( $P$ 、 $Q$  とする) を探索する。次に、 $P$  から光源に向かってレイを生成し、雲が存在するすべてのボクセルとの交点のうち最も  $P$  から遠いボクセル ( $R$  とする) を得る。最後に、 $R$  から  $Q$  に向かう方向を算出し、それを新たな直射光の方向とする。これにより、ユーザは画面上で直接陰影部分を選択し、それを所望の方向へ動かすだけで、直感的に陰影を動かすことが可能である。

## 4. 実験結果

提案手法を適用した結果を図4に示す。実験環境は、CPUがIntel Core 2 Quad Q9400、GPUがNVIDIA GeForce GTS 250、グラフィックスAPIがOpenGLである。また、GPU

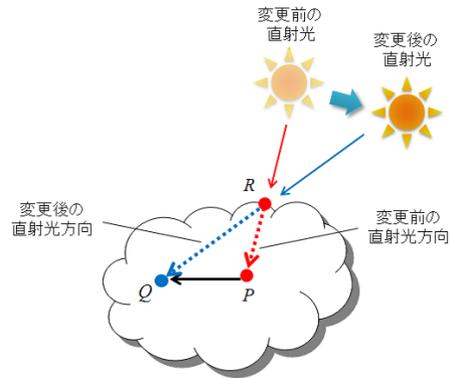


図3 陰影の移動  
Fig.3 Moving shadows

プログラミング言語として CUDA を用いて計算を行っている。図4(a)は直射光の色を設定するためにペイントしたストロークを示す。図4(b)は、その結果設定された直射光を用いてレンダリングした画像である。ユーザが色塗りを行った部分の色が、編集後のレンダリング結果の該当部分の色と等しいことが確認できる。次に、図4(c)は、3.3節で述べた方法を用いて、直射光方向を設定した結果である。ユーザは、図4(b)に対して移動させたい陰影部分を画面上で直接選択し、それを所望の位置へ移動させることで、容易にこのような結果を生成することができる。図4(c)の結果に対し、視点位置を変更して図4(d)のような色塗りを行い、環境光成分を設定した結果が図4(e)である。ユーザの指定した色情報が周囲に伝播していることと、ボクセルごとに異なる環境光成分が割り付けられていることが確認できる。図4(e)に対して、視点位置を変更した結果が図4(f)である。この結果からも、ユーザの指定した色情報が自然に周囲に伝播していることが確認できる。図5も、同様な処理によって、環境光の設定を行った例であるが、この例では、環境光成分に青色を設定している。

次に、直射光の色に関する結果の比較を図6に示す。図6(a)のような色塗りを行い、提案法を用いて直射光の色を設定した結果が図6(b)である。一方、提案法を用いず、図6(a)で指定している色をそのまま直射光の色として設定した結果が図6(c)である。図6(c)では、ユーザの指定した色とレンダリング後の色に差が見られる。そのため、レンダリ

ング後の色を容易に予想することができる提案法の方がより直感的であると言える。また、環境光成分に関する結果の比較を図7に示す。図7(a)に対して、提案法を用いて陰影部分を明るくするように環境光成分を設定した結果が図7(b)である。提案法により、陰影部分のみが明るくなっていることが確認できる。一方、提案法を用いず、陰影部分を明るくするように環境光成分を設定した結果が図7(c)である。提案法を用いない場合、全ボクセルに対して一定の環境光成分が割り付けられるため、陰影以外の部分についても輝度が大きく変化していることが確認できる。

## 5. ま と め

本稿では、ユーザの編集操作に応じて雲の輝度計算式の各パラメータを変更することで、雲のボリュームレンダリングを編集することができるシステムを提案した。提案法には直射光の色を設定するモード、環境光成分を設定するモード、直射光方向を設定するモードの三つの編集モードが用意されており、それぞれのモードを用途に応じて使い分けることで、ユーザは直感的に所望のボリュームレンダリング結果を生成することが可能となった。今後の課題としては、新たな編集モードを実装することが挙げられる。特に、リアルな雲のレンダリングの重要な要素の一つである多重散乱光に関する編集モードを実装し、簡単にリアルなレンダリング結果を生成できるようシステムを拡張したい。

## 参 考 文 献

- 1) T. Ritschel, M. Okabe, T. Thormahlen and H.P. Seidel : Interactive Reflection Editing, *In Proceeding of ACM SIGGRAPH Asia*, Article No. 129 (2009).
- 2) T. Ritschel, T. Thormahlen, C. Dachsbacher, J. Kautz and H.P. Seidel : Interactive On-Surface Signal Deformation, *In Proceeding of ACM SIGGRAPH*, Article No. 36 (2010).
- 3) F. Pellacini : envyLight : An Interface for Editing Natural Illumination, *In Proceeding of ACM SIGGRAPH*, Article No. 34 (2010).
- 4) A. Levin, D. Lischinski and Y. Weiss : Colorization using optimization, *In Proceeding of ACM SIGGRAPH*, pp.689-694 (2004).
- 5) William H. Press, Brian P. Flannery, Saul A. Teukolsky and William T. Vetterling : Numerical Recipes in C Second Edition, Cambridge University Press, New York (1973).

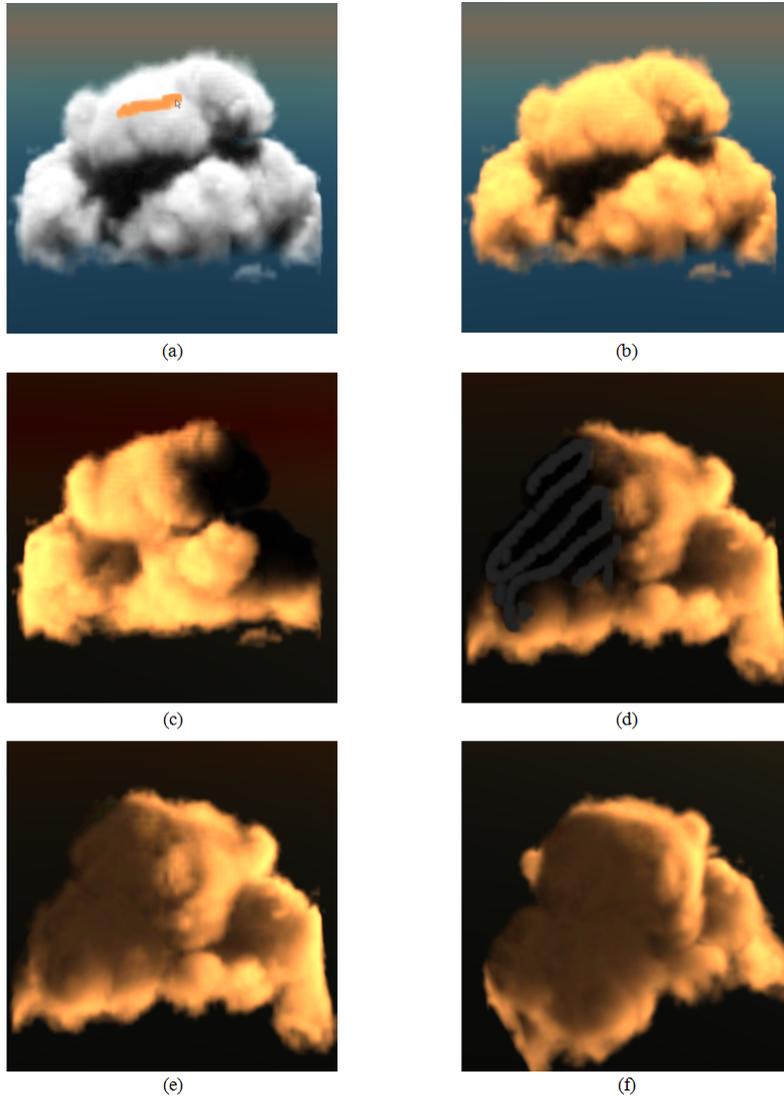


図 4 提案法を適用した結果  
Fig. 4 Results using the proposed method

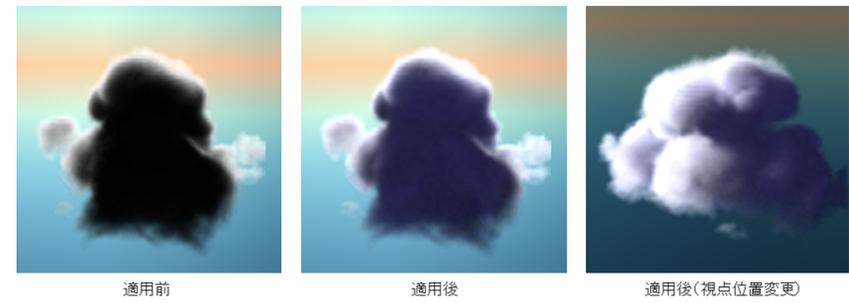


図 5 その他の結果  
Fig. 5 Another example

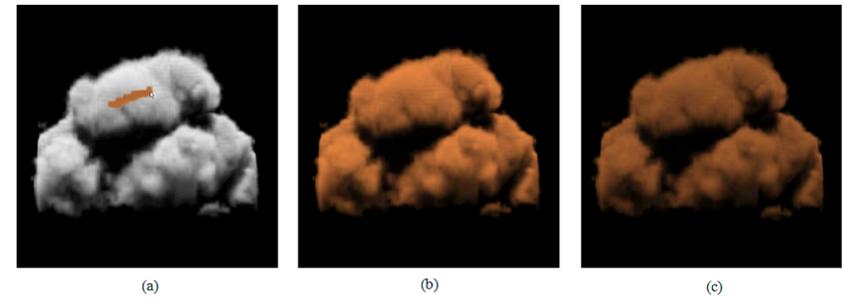


図 6 直射光の色に関する結果の比較  
Fig. 6 Comparison of effects of the direct light

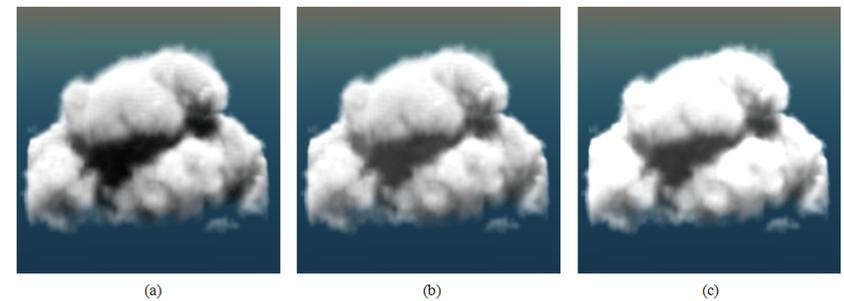


図 7 環境光成分に関する結果の比較  
Fig. 7 Comparison of effects of ambient light