

画像データベースを利用した空の編集システム

小野 亜由美^{†1} 土橋 宜典^{†1} 山本 強^{†1}

本研究では、屋外景観を構成する重要な要素である空に着目し、ユーザの意図を反映できる直感的な空の画像の編集システムを開発する。提案手法では、まず空の輝度分布と色を考慮してインタラクティブに空を生成する。そして雲の写真データベースから所望の雲を選択して合成する。しかし、合成した雲の種類や影の付き方によっては不自然な結果になってしまうことがある。そこで、作成した空に合う雲の写真をデータベースから検索し、ユーザに提示するシステムを提案する。

An Editing System for Sky Using Photographic Database

AYUMI ONO,^{†1} YOSHINORI DOBASHI^{†1}
and TSUYOSHI YAMAMOTO^{†1}

In this research, we focus on the sky that is an important element of outdoor scenes, and develop an intuitive editing system for sky image that can reflect the user's intension. Our method consists of two steps. The first step is a sky generation process, and the second step is a cloud synthesis process. In the first step, the intensity distribution and colors of the sky are created by using our intuitive user-interfaces. In the second step, we compose images of cloud extracted from photographs onto the sky created in the first step. However, the result might be look unnatural if the cloud types or shadows are inconsistent with the created sky. So, we propose a system that searches for cloud photographs that matches to the created sky.

1. はじめに

屋外で撮影された写真や、コンピュータグラフィックスで生成された屋外シーンには、多

くの場合背景として空が含まれている。空は様々な色をしており、空に浮かぶ雲も非常に多様である。この空を背景として効果的に利用することによって、そのシーンに様々な印象を与えることができる。このようなことから、空は屋外シーンを構成する重要な要素であるといえる。そのため、コンピュータグラフィックスを用いて空を生成する手法は多く存在する。例えば、リアルな空を生成する手法としては、物理シミュレーションによる手法が挙げられる。物理計算に基づいているため、任意方向の空の色を高精度に算出することができる。しかし、物理シミュレーションは多くのパラメータに基づいており、必ずしもユーザの希望通りの空を作成することは限らない。また、大量の空の写真のデータベースから、所望の写真を効率的に検索する手法も存在する。この手法では、直感的に空画像の検索を行える。しかし、必ずしも所望する空の画像がデータベースに含まれているとは限らない。そこで、本研究ではユーザが直感的に所望する空を作成することのできるシステムの開発を行う。提案手法では、空の輝度分布、空の色、雲の3つについて考慮する。まず、空の輝度分布と色についてユーザがマウス操作によって直感的に編集を行い、所望の空を作成する。そして作成した空に対して実写の雲を合成する。雲については、雲の写真データベースから作成した空に合う雲の写真を検索してユーザに提示する。ユーザはその中から所望の雲の写真を選択し、合成を行う。このようにして、ユーザの希望に沿った自然な空を作成することが可能となる。

以降、本論文の構成は、2節で関連研究についての簡単な解説を行い、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装結果とその考察について述べ、最後に5節でまとめとする。

2. 関連研究

空を表示する手法としては、物理シミュレーションによる手法1) 2) が挙げられる。文献1)では、大気分子によるレイリー散乱などを考慮した空の色の計算を高速に行うことができる。物理法則に基づいて計算を行っているため、リアルな空の色が算出される。しかし、この手法は多くのパラメータに依存しているため、パラメータを調整してユーザの所望する空を作成するのは困難である。文献2)では、物理シミュレーションを利用し、空の輝度分布を解析的な式で表現している。必要なパラメータはユーザが容易に測定・推定可能なものとなっているが、この手法もパラメータを調整してユーザの希望を反映させるのは困難である。

また、空の写真を検索する手法として文献3)が挙げられる。これは、ユーザが空の種類

^{†1} 北海道大学大学院情報科学研究科

Graduate School of Information Science and Technology, Hokkaido University

やレイアウト、太陽位置などの属性を指定することにより、所望する空の写真を検索するシステムである。現在主流となっているテキストに基づく画像検索では、手動で入力されたキーワードや、画像周辺にあるテキスト情報を用いて検索が行われている。それに対し、この手法ではあらかじめ空の写真の属性を自動的に抽出してタグ付けしておくことにより、内容を考慮した検索が可能である。しかし、必ずしもユーザの所望する空の画像がデータベースに含まれているとは限らない。

これらのことを踏まえ、本研究では直感的な空の作成を目的とする。

3. 提案手法

提案手法の概要を図1に示す。まず空のモデルを用いて空をグレースケールの全周画像として表現し、空の輝度分布の編集を行う。編集は全周画像上のマウス操作で簡単に行うことができる。そしてその全周画像に対してユーザが部分的に所望の色を指定し、空全体の色付けを行う。このようにして作成した空の全周画像から、任意方向の空画像を生成する。次に、実写の雲の合成を行う。しかし、雲の種類や影の位置が生成した空画像と合わないものを合成してしまったときに不自然な結果になってしまうことがある。そこで、提案手法では作成した空とデータベース内の写真の空領域の類似度を算出し、類似度が高い写真をユーザに提示する。ユーザはそこから所望の写真を選択し、作成した空画像の任意の位置に合成する。それぞれの処理について、以下で詳しく説明する。

3.1 空の輝度分布の編集

空の輝度分布は、文献4)のモデルを用いて表現する。このモデルは、実測値の近似式であり、任意方向の空の輝度 L は次式で与えられる。

$$L = L_z \frac{f(\theta_p, \gamma)}{f(0, \theta_s)} \quad (1)$$

$$f(x, y) = \{1 + a \exp(b/\cos x)\} \{1 + c \exp(dy) + e \cos^2 y\} \quad (2)$$

ここで、図2に示すように、 θ_p は天空の注目方向の天頂角、 θ_s は太陽方向の天頂角、 γ は注目方向と太陽方向のなす角を表している。また、 L_z は天頂輝度、 a, b, c, d, e は気象係数と呼ばれる定数である。気象係数はそれぞれ以下のものを表している。

- a : 天頂角に対する輝度勾配
- b : 地平線付近の輝度勾配
- c : 太陽周辺領域の輝度
- d : 太陽周辺領域の大きさ

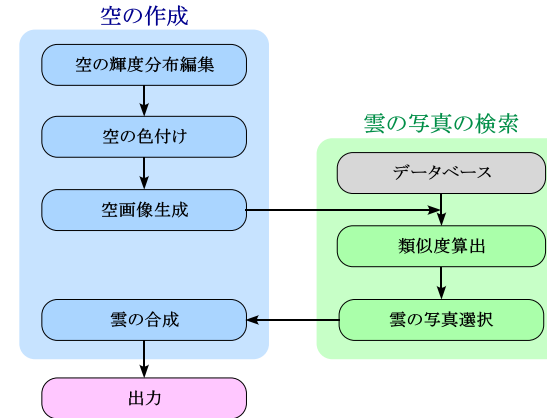


図1 提案手法の概要
Fig. 1 Outline of the proposed method

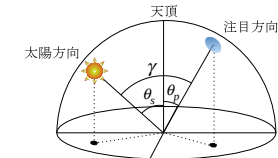


図2 Perez sky model
Fig. 2 Perez sky model

e : 地面における後方散乱光の強度

これを RGB の3つのカラーチャネルについて考慮することで、カラーの空の輝度分布を表現することが可能である。しかし、その場合はパラメータ数は18個となってしまい、それらを全て調整することは極めて煩雑であり、ユーザの意図を反映させるにくくなってしまふ。そこで、まず本手法ではグレースケールで空の輝度分布を作成した後、色付けを行う。色付けについては、3.2節で述べる方法により行う。ここでは、輝度分布の編集方法について述べる。

本手法では、半球の空を仰角および方位角についてサンプリングし、全周画像として表示。各パラメータの調整は、この全周画像上でのマウス操作により行う。太陽位置は、太陽の天頂角および方位角をマウス位置に一致させることで、図3のようにドラッグにより移動させる。太陽領域の輝度および大きさは、気象係数 c および d を太陽の中心とマウス位置の距離に比例させた値にすることによって行う。輝度勾配は、気象係数 a, b および e によって調節する。上方向にドラッグすると輝度勾配が大きくなり、下方向にドラッグすると輝度勾配が小さくなる。特に地平線付近でドラッグを行うと、地平線付近の輝度勾配が変化するようにになっている。このように、マウス操作で簡単にかつインタラクティブに空の輝度分布調整し、所望の結果を得ることができる。

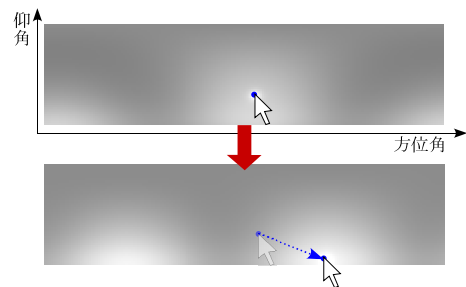


図3 太陽位置の移動
Fig. 3 Moving sun position

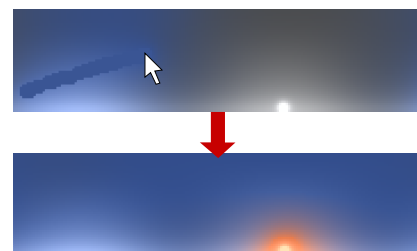


図4 空の色付け
Fig. 4 Colorization of the sky

3.2 空の色付け

3.1 節の方法により作成したグレースケールの空に対し、色付けを行う。本手法では、直感的に色付けをすることに重点を置き、文献 5) の手法を利用する。これは、グレースケールの画像に対してユーザが部分的に色情報（色ストローク）を指定することにより、画像全体をカラー化する手法である。色ストロークの指定は、ペイントソフトの要領で画像の所望の位置に所望の色を塗るだけでよく、非常に簡単に行うことが出来る。以下で、この手法の詳細について説明する。

この手法では YUV 色空間を採用しており、Y は輝度、U は輝度と青の色差、V は輝度と赤の色差を表している。入力はグレースケール画像とユーザが指定した色情報であるため、Y については全ピクセル、U および V についてはユーザが色ストロークを指定したピクセルにおいて既知である。これらの情報から全ピクセルの U および V を求めることで、画像全体がカラー化される。

各ピクセルの U は、以下の式 (3) の $J(U)$ を最小化することにより求められる。

$$J(U) = \sum_r \{U(r) - \sum_s w_{rs} U(s)\}^2 \quad (3)$$

ここで、 r は注目ピクセル、 s は注目ピクセルの 8 近傍ピクセルであり、 $U(r)$ 、 $U(s)$ はそれぞれ r 、 s における U である。また、 w_{rs} は式 (4) で表される。

$$w_{rs} \propto \exp \left(-\frac{(Y(r) - Y(s))^2}{2\sigma_r^2} \right) \quad (4)$$

画像のピクセル数を n とすると、式 (3) の最小化問題は n 元の連立一次方程式に帰着される。これを解くことによって注目ピクセルの U が求められる。また、V についても同様に



図5 空の写真
Fig. 5 Photographs of the sky



図6 データベースからの検索
Fig. 6 Searching sky images from database

求めることができる。

本手法では、連立一次方程式の解法として Jacobi 法を採用し、GPU を用いて並列計算を行うことで高速化を行った。また、更なる高速化を図るために、2 段階の解像度でのカラー化を行った。まず初めに、入力画像をダウンサンプリングして上述のカラー化手法を適用する。低解像度であるため、前述の連立一次方程式の解は高速に収束する。そして、その結果をアップサンプリングして得られた U および V を、本来の解像度でのカラー化の初期値とする。初期値が既に解に近い値となっているため、解を高速に収束させることができる。このように、GPU による実装と適切な初期値の決定を行うことでリアルタイム処理が可能となり、図 4 のように色ストロークを指定している間もカラー化を行うことが可能となる。そのため、結果を確認しながら色塗りを行うことができ、インタラクティブかつ直感的な色付けを実現することができた。このようにして生成した空の全周画像を用いると、任意方向の空画像を得ることができる。

3.3 雲の写真の検索

3.2 節で作成した空画像に、実写の雲を合成する。しかし、例えば夕焼け空に青空の雲を合成した場合などに、空と雲の色や陰影が合わず不自然になってしまうことがある。そこで、作成した空画像と空が類似している写真をデータベースから検索し、その写真中の雲を

合成するという方法を用いる。空の写真は、Flickr.com(<http://www.flickr.com/>) のグループから「青空」や「夕焼け」などのキーワードを用いてダウンロードした。図 5 はその一部である。また、それぞれの写真を 64×64 に縮小したサムネイル画像も作成して保存しておく。以後の計算にはこのサムネイル画像を利用し、計算コストの削減を図る。

3.3.1 空領域の特定

作成した空画像とデータベース内の空画像の類似度は、空領域の色を利用して算出する。そのため、まず写真中の空領域を示すマスクを作成する必要がある。空領域は、全体から「黒い領域」および「青空中の雲領域」、「色の平均値から大きく外れている領域」を除外した領域とする。黒い領域は、RGB 色空間における閾値判定により特定する。黒い領域を除くことで、夕焼けの写真の前景部分等を除くことができる。青空中の雲領域の特定は、まず青みの強いピクセルが多い写真を青空と判定し、青空と判定された場合にのみ文献 7) の彩度による閾値判定を適用して雲領域を特定する。ただし、雲領域が全体の 80 % 以上となった場合は、80 % 以下になるまで閾値を少しずつ下げながら再判定を行う。その後、空領域と考えられている領域内の色の平均値を算出し、平均値から大きく外れているピクセルを空領域から除外する。このようにして得られた領域を空領域とする。

3.3.2 画像間の類似度

空画像の特徴として、水平方向よりも垂直方向について、輝度に変化する傾向がある。そこで、図 7 のように画像を垂直方向について N 個の小領域に分割し、画像 X の n ($n \leq N$) 番目の小領域を X_n と表す。画像 X, Y 間の類似度 $S(X, Y)$ を、以下の式で定義する。

$$S(X, Y) = \frac{1}{\sum_i \sum_{n=1}^N |avg(X_n) - avg(Y_n)|} \quad (5)$$

ここで、 i はカラーチャネル数、 avg は色の平均値を算出する関数である。この式を用いて、作成した空画像 I とデータベース内の空の写真 D の類似度を算出する。ただし、ここでは空領域の類似度が必要なため、式 (5) の avg の計算は空領域のみで行う。また、色空間には人間の視覚を近似するよう設計されている $L^*a^*b^*$ 色空間を利用する。

データベース内の写真 D については、あらかじめ $avg(D_n) \{n = 1, 2, \dots, N\}$ を計算し、タグ付けしておく。これにより、高速に類似度の算出を行なうことができる。類似度の算出が完了すると、類似度の高い画像を複数枚ユーザに提示する。

3.4 雲の合成

3.3 節の検索結果から所望の写真を選択し、作成した空画像にその写真の雲を合成する。合成には、文献 6) の手法を利用する。これは、選択した領域の色を貼り付け先の画像の色

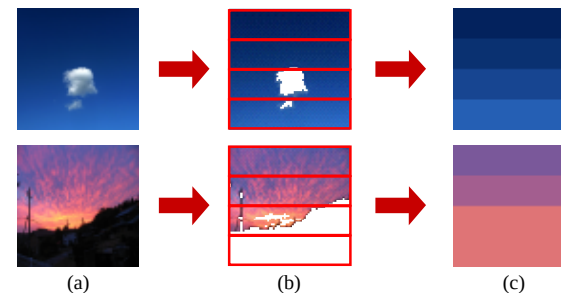


図 7 空領域の色の平均 ($N=4$)
Fig. 7 Averages of sky region ($N=4$)

に馴染ませ、シームレスな合成を行なう手法である。以下でこの手法について簡単に説明する。

図 8 に示すように、選択した領域の輝度を g 、輝度勾配を $\mathbf{v} = \nabla g$ とする。また、貼り付け先の画像 S の貼り付ける領域 Ω の輝度を f 、それ以外の領域の輝度を f^* とする。 f を求めるためには、境界条件を $f|_{\partial\Omega} = f^*|_{\partial\Omega}$ とし、式 (6) を解く。

$$\nabla f = \nabla g \quad (6)$$

すなわち、境界線上では貼り付け先の画像に輝度を利用し、境界線の内部では選択した画像の輝度勾配を用いることとなる。式 (6) はガウス・ザイデル法などの反復計算によって数値的に解くことができる。

なお、合成する領域の選択は以下の 2 種類のモードによりユーザが行う。1 つ目のモードでは、ユーザが任意の箇所をクリックすると、クリックした点のピクセルと色が似ている領域も自動的に選択される。そして 2 つ目のモードでは、ペイントソフトの要領で選択したい領域を塗っていく。また、合成位置はユーザがマウス操作で所望の位置を選択することができる。

Poisson Image Editing の問題点として、境界線上 $\partial\Omega$ で、選択した領域の輝度 g と貼り付け先の輝度 f^* の差が大きいと、算出した f が g と大幅に異なり、望ましくない結果となってしまうことが挙げられる。本手法の場合は、空が似ている画像から合成を行うため、色が大幅に異なることは少ない。しかし、領域を指定する際に雲とそれ以外の領域をまたいでしまうと、境界での色の差が大きくなってしまふ。そこで、図 9 に示すように、境界の外側を背景の空の色で囲んで境界線の色を修正してから計算を行うようにした。

また、Poisson Image Editing では色は自然に馴染むが、輝度勾配はそのまま残るため、

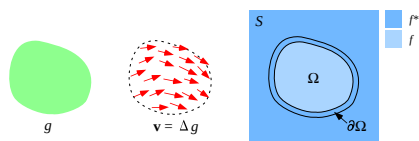


図 8 Poisson image editing
Fig.8 Poisson image editing

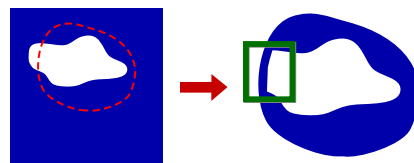


図 9 境界の色の修正
Fig.9 Boundary color adjustment

境界が目立ってしまうことがある。そこで、合成する領域に対して距離変換を行い、境界線に近いピクセルはぼかしながら合成を行った。

4. 実装結果

提案法による実装結果を示す。実装環境は、CPU:Core2 Quad CPU Q9400, GPU:GeForce GTS 250, グラフィックス API:OpenGL である。なお、GPU プログラミング言語として CUDA を用いている。

まず、青空を目標に空の作成を行った結果を図 10 に示す。図 10(a) は作成した空の輝度分布、図 10(b) は指定した色ストローク、図 10(c) はカラー化結果を示している。ユーザの希望を反映させた空が作成できていることが確認できる。

次に、データベースから写真の検索を行った結果と、検索された写真を用いて雲の合成を行った結果を図 11 に示す。まず、図 11(a) は全周画像から作成した空画像である。そして、図 11(b) は図 11(a) と空が似ている写真を検索した結果の上位 15 枚である。似た画像が検索出来ていることがわかる。なお、検索に要した時間はおよそ 0.801 秒であった。そして、図 11(b) の赤枠の写真を用いて雲の合成を行った結果が、図 11(c) である。自然な合成が出来ていることが確認できる。

図 12 はその他の結果を示している。このように、様々な空の画像を作成することが可能である。

5. ま と め

輝度分布と色に関してユーザの意図を反映させた空画像を作成する手法を提案した。また、作成した空画像に合う実写の雲を検索し、合成する手法を提案した。提案法により、ユーザの希望を反映させた空を直感的に作成することが可能となった。しかし、現在は雲の写真を検索する際に、写真中の空領域の色にしか着目していない。そこで、太陽位置等についても

考慮し、検索精度の向上を図る必要がある。

参 考 文 献

- 1) Y. Dobashi, T. Nishita, K. Kaneda and H. Yamashita: A Fast Display Method of Sky Color Using Basis Functions, *Proc. Pacific Graphics'94*, pp.194–208, Beijing, China, Aug. (1994).
- 2) A.J. Preetham, P. Shirley and B. Smits: A Practical Analytic Model for Daylight, *Proc. SIGGRAPH 99*, pp.91–100, Los Angeles, "California USA, Aug. (1999).
- 3) L. Tao, L. Yuan, and J. Sun: SkyFinder : Attribute-based Sky Image Search, *Proc. SIGGRAPH2009*, New-Orleans, Louisiana, USA, Aug. (2009).
- 4) R. Perez, R. Seals and J. Michalsky: All-weather Model for Sky Luminance Distribution-preliminary Configuration and Validation, *Solar Energy*, vol.50, no.3, pp.235-245, March (1993).
- 5) A. Levin, D. Lischinski and Y. Weiss: Colorization Using Optimization, *Proc. SIGGRAPH 2004*, Los Angeles, California, USA, Aug. (2004).
- 6) P. Perez, M. Gangnet and A. Blake: Poisson Image Editing *Proc. SIGGRAPH 2003*, vol.22, no.3, pp.313-318, (2003).
- 7) 高野松美: 雲監視装置による全雲量算出方法とその結果, 高層気象台彙報第 67 号, pp. 67-74 (2007).

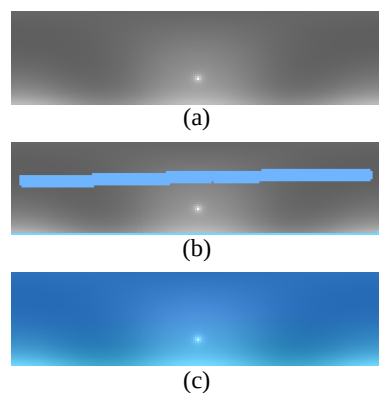


図 10 空の全周画像

Fig. 10 An image of the whole sky

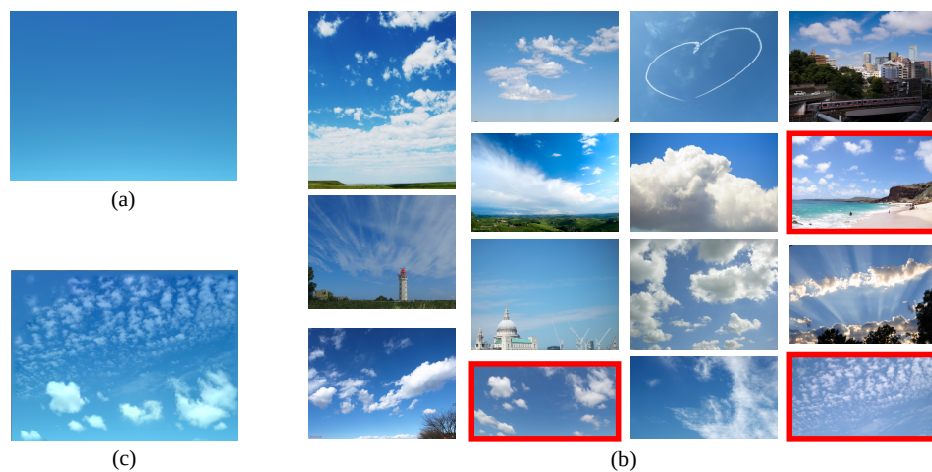


図 11 検索結果と雲の合成結果

Fig. 11 Search and cloud composition result

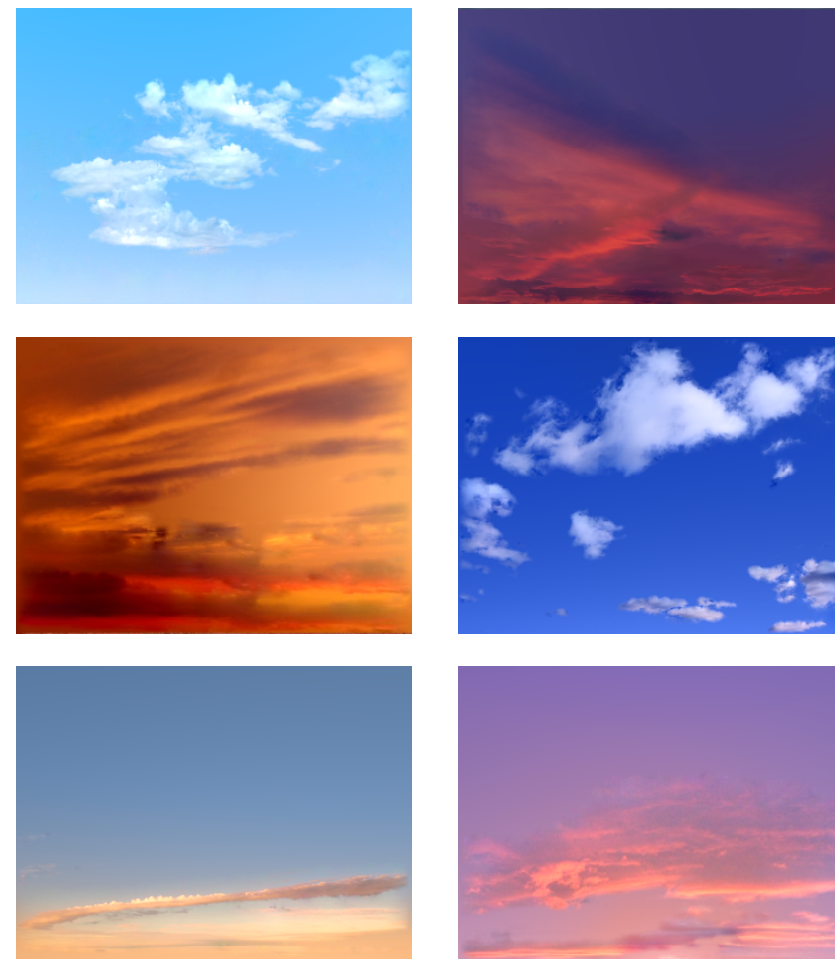


図 12 その他の結果

Fig. 12 Other examples