

環境マップのインタラクティブなデザイン手法

岡部 誠
東京大学

松下 康之
Microsoft Research Asia

五十嵐 健夫
東京大学/JST PRESTO

3次元シーンのデザイン支援の一環として、我々は image-based lighting における環境マップをデザインするための、簡単で使いやすいユーザ・インタフェースを提案する。提案システムを用いることで、デザイナーは3次元シーンに直接ペイント操作を行うことで、シーンの最終的なレンダリング結果を調整しながら環境マップをデザインすることができる。3次元シーンを、デザイナーが意図したようにレンダリングするための環境マップは、inverse lighting 問題を解くことによって求められる。本手法が、3次元シーンの照明デザイン作業に対して、有用でかつ使いやすいことを示す。また、本手法が3次元モデルを写真へ挿入する作業において有用なアプリケーションであることも示す。

Interactive Design of Environment Map

Makoto Okabe
Tokyo Univ.

Yasuyuki Matsushita
Microsoft Research Asia

Takeo Igarashi
Tokyo Univ./JST PRESTO

To help artists design customized lighting environments, we present a simple user interface for designing image-based lighting. Our system allows the artist to directly specify the appearance of the resulting image by painting and dragging operations. Our system constructs an image-based lighting model that produces the desired (and painted) appearance by solving the inverse shading problem. We also provide an interface with which the artist can drag highlights and shadows on the surface to control the orientation of the entire lighting environment. We demonstrate that our technique is useful and practical for adding lighting effects to synthetic objects. We also show an application of seamlessly merging synthetic objects into photographs.

1. はじめに

Image-based lighting、もしくは環境マップと呼ばれる技術によって、シーン周囲の照明環境をテクスチャ画像としてコンパクトに表現することができる。この手法によって、無限遠から届く複雑な入射光がコンパクトに表現でき、またリアルタイムなレンダリングも可能になっている[3]。しかし、ほとんどのシステムは、実世界を撮影した環境マップに依存しており、手作業で環境マップをデザインしようという試みはほとんど行われていない。デザイナーは環境マップを直接ペイントすることができるが、この作業は難しく、大変な労力が必要である。この原因は、3次元モデルの特定の場所を光らせるために、環境マップ上のどこを塗ればよいのか、という関係性が分かりにくいことに起因する。しかし、デザイナーの望む環境マップが、実世界を撮影することによって得られることは保証されており、従って、環境マップをマニュアルでデザインするためのツールが必要とされる。

この問題を解決するために、我々は見た目を直接操作するユーザ・インタフェースを提案し、環境マップデザインを支援する。ライトを3次元空間に配置してからレンダリング結果を確認するのではなく、ユーザは3次元モデルの表面に直接色を塗ったり、その色をドラッグしたりといった操作を行う。それに伴い、システムは inverse lighting 問題を解くことで、環境マップをデザインする。ライトデザインのための見た目に基づくユーザ・インタフェースは過去にも提案されている[11,15,12,6,8,1,17]が、全て点光源や並行光源などの単純なライトモデルに特化したものであり、写実的なレ

ンダリングを可能にする環境マップをデザインするためのものは存在しない。とくに、環境マップは、実写画像もしくは実写映像に3次元モデルを合成する際、光の調整に便利である。

ユーザが3次元モデルの見た目を指定すると、システムはその見た目を作するような環境ライトを計算する。この処理は inverse lighting 問題として定式化される。この問題は、シーンの幾何形状、反射している光の強さ、物質情報(BRDF)の3つが既知であるとして、入射光を求める問題である[13]。この問題をインタラクティブに解くために、我々は spherical radial basis function(SRBF)を利用した precomputed radiance transfer(PRT)[18]を用い、環境マップを SRBF で近似する。Inverse lighting 問題はこの SRBF の係数を推定する問題に置き換えることができる。SRBF に基づく PRT を導入することで、本手法は分析的な BRDF と計測に基づく BRDF の両方を扱うことができる。システムはインタラクティブな視点変更とライトの更新を実現している。

2. 既存研究

光源パラメータを手動で設定する以外の方法として、3次元シーン内で照明効果を操作する方法は過去にもいくつか提案されている。Pellacini らはキャストシャドウを使用した照明デザイン手法を提案した[11]。また他にもスケッチやペイント操作により直接照明効果を描く手法が提案されている[15,12,6,17]。これらの手法は効率的であるが、単純な局所照明モデルや単純な形状に限定されている。他の関連研究としては心理物

理学に基づく手法[8]、トゥーンレンダリング下での照明効果のデザイン[1]などがある。また自動化された光源デザインシステムは[16,4]でも提案されている。

我々のシステムは inverse lighting 問題を扱っている。この問題は、シーンの形状、反射特性、レンダリング結果がわかっているときに、光源の分布を推定する問題である。既存の手法では光源分布をいくつかの異なる入力、影[14]、陰影[19]などから推定していた。Liらはさまざまな手がかりを駆使し光を推定している[9]。Nishinoらは人間の目に映った映像を元に光源推定を行っている[10]。また Ramamoorthiらは信号処理に基づいたBRDFと光源の推定フレームワークを提案している[13]。我々の手法はこれらの技術の上に成り立つものである。

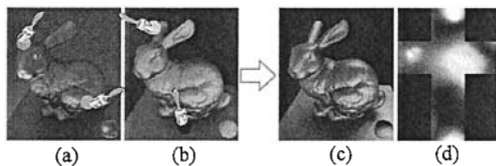


図1：イルミネーションブラシによる環境マップのデザイン。バニーのモデルはアルミニウム・ブロンズを測定したBRDFにより表現されている。(a)ユーザはオレンジ色と緑色を指定し、バニーのディフューズな見た目を指定している。(b)シーンは推定された環境マップによりレンダリングされた。ユーザはさらに水色のハイライトと暗い影を塗っている。(c)指定された全ての照明効果が満たされるような環境マップが推定された。結果の環境マップを(d)に示す。

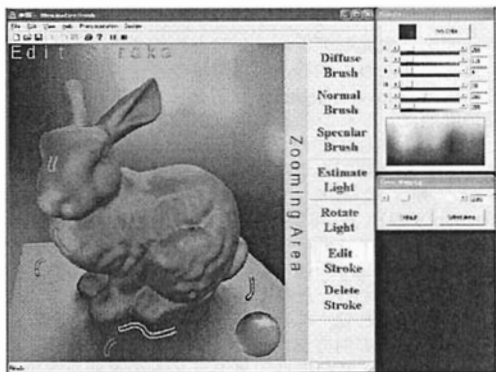


図2：提案システムのスナップショット

3. ユーザ・インタフェース

この章では、デザイナーの視点に立って、照明デザインシステムについて説明する。本システムを使うことで、ユーザは直接、シーンの見た目をスクリーンに描き込むことができる。システムはまた、照明効果を修正するための、ドラッグツールもサポートしている。環境マップとスクリーン画像はインタラクティブに更新される。ユーザは自由に視点の変更をしながらライトのデザインを行うことができる。全ての光源は無限

遠にあると仮定されている。また、色を塗る操作は、ライトの位置、色、強さ、形状に影響を及ぼす。本手法はシーンの形状とBRDFが既知であるものとしている。図2は提案システムのスナップショットである。新しい3次元シーンを与えられると、ユーザはまず数分間の前計算を行い、それから、インタラクティブな照明デザインを開始する。

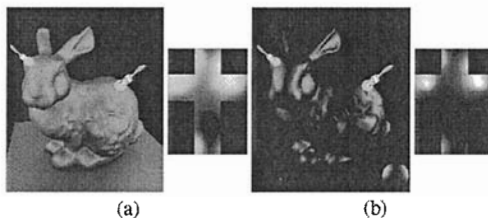


図3：イルミネーションブラシの効果。バニーモデルは測定によって得られたアルミニウム・ブロンズBRDFを持っている。(a)ユーザは緑と紫の色を、ディフューズブラシを用いて3次元モデルにペイントしている。結果の環境マップからは拡散した光源が生成されている様子がわかる。(b)では、ユーザは同じ色を同じ位置にスペキュラーブラシを用いてペイントしている。結果の環境マップはよりシャープな構造を持っており、モデルの表面でもスペキュラーハイライトが見られる。

3.1. イルミネーションブラシ

イルミネーションブラシはサーフェス上に望まれる見た目を指定するためのペイントツールである。ユーザはドラッグ操作により、画面内にストロークを描画することができる。ストロークは物体表面に投影され、近傍の頂点に色の制約を与える。この制約をもとに光源が推定される。色と位置は、あとで変更することもできる。システムは2種類のブラシを提供している。1つはディフューズ成分を塗るための、ディフューズブラシ、もう1つがスペキュラー成分を指定するためのスペキュラーブラシである。例を図3に示す。スペキュラーハイライトが視点の変化に依存して物体表面上をキラキラ動くのに対し、ディフューズ成分は視点に無関係な効果を言う。どちらのブラシを使う場合も、ユーザは色を選んで3次元モデルを塗る、という単純な操作を行う。ペイント操作が行われるたびに、システムは即座に環境マップの推定、更新を行う。

しかし、ユーザのインプットというのは必ずしも一貫性があるものではない。例を図4に示す。(a)では、ユーザは、白と黒を異なる2つの位置にペイントしている。ただし、この2つの位置で、法線の向きは同じである。すると、この入力を完全に満たす照明というのは存在せず、システムは最小二乗法によりもっともユーザの入力を満たす解を計算し、(b)では、塗られた位置が同じグレー色を持つ結果が得られている。このグレー色とは、白と黒の平均値である。一方、(c)でもユーザが再び、白と黒を同じ法線の地面に塗っている。しかしこの場合、黒で塗られた箇所は像の影になることができるため、この2つの入力は一貫性があり、システムも照明環境をユーザの意図に沿って推定することができる。結果として(d)に示すように、ソフトシャドウが生成された。

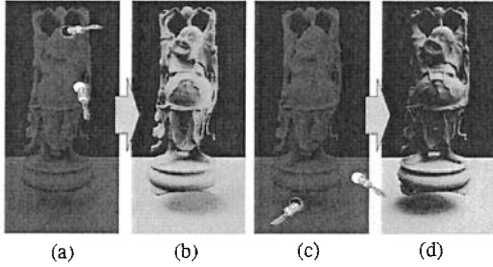


図4：(a-b)一貫性のないペイント入力。(c-d)一貫性のあるペイント入力。前者は入力された色が平均された照明効果が得られ、後者はソフトシャドウが得られる。

3.2. 環境マップの回転

照明デザインの最中に、全体的な環境マップの回転操作が行うことができる。我々は、この目的のためにも、シーンの見た目を直接操作できるツールを提供する。ユーザはシーン中のあらゆる特徴、たとえばハイライトや影などをクリック、ドラッグして、環境マップを回転させることができる(図5)。内部的には、システムは環境マップを回転させながら、マウスの下にある色が一定に保たれるようにしている。

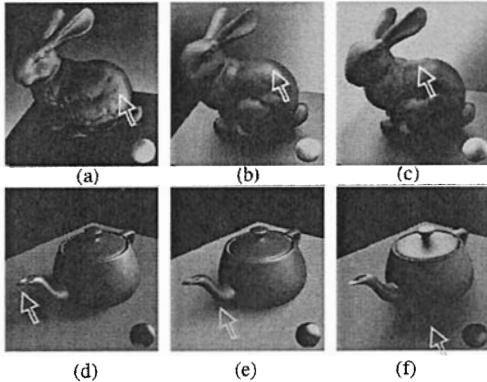


図5：(a-c)ユーザは黄色く明るい部分をつまんでドラッグしている。それにつれて、環境ライトがスムーズに回転している。(d-f)影をドラッグしながら、環境マップを回転させることもできる。

4. アルゴリズム

この章では、インタラクティブなレンダリング手法とユーザ・インタフェースのアルゴリズムについて述べる。インタラクティブなレンダリングアルゴリズムを開発するにあたって、我々は Tsai らによって提案された SRBF に基づく PRT フレームワークを応用する[18]。理由を以下に述べる。まず、この手法はレンダリングスピードが速く、ユーザにレンダリング結果をリアルタイムにプレビューさせることができる。2つ目の理由は、spherical harmonics や wavelet を使用した手法とは異なり、基底関数が周波数空間ではなく、空間領域に存在していることである。これは推定アルゴリズムを単純に、そして安定にしている。

レンダリングアルゴリズムは既存の PRT 手法に基づいている[18]：システムは、シーンの形状、BRDF を固定し、視点とライトのリアルタイムな変更を可能にしている。この手法の大きな問題点は前計算にかかる時間である。たとえば、50,000 頂点のシーンに対し、照明デザインをするには、24 時間の前計算が必要となる。それゆえ、我々のシステムは計算結果を、単純なモデルを用いて近似し、前計算にかかる時間を短縮する。まず、Phong のモデルと同様な、三角形内部の色補完を、視線ベクトルに基づいて行う方法をやめ、三角形内部の色補完は、3 頂点の持つ色を単純に補完することにする。次に、SRBF による関数近似する際に、単純な手法を使用する。既存手法のようにグローバルに最適な SRBF の係数を最小二乗法で計算するのではなく、我々は各々の SRBF が比較的独立であると仮定して、係数の計算を独立に行う。これらの変更点は前計算にかかる時間を 1 日から数分に減らすことができる。これらの近時はまた、レンダリングの正確さとスピードにおいて、既存手法よりも劣っているが、照明デザインという目的に使うためには十分である。

4.1. インタラクティブ・レンダリング

無限遠の照明に関しては、あるサーフェス上の点 x から放射される光の強さは、レンダリング方程式[7]を用いて記述される：

$$I(x, \omega_o) = \int_{\Omega_i} L(\omega_i) R(x, \omega_i, \omega_o) V(x, \omega_i) \max(0, \omega_i \cdot n) d\omega_i, \quad (1)$$

ここで、 ω_i と ω_o は光と視線の方向である。 L は環境マップからの入射光、 R は BRDF、 V は隠蔽マップ(0 か 1 の値を取る)、 n は法線である。式(1)を書き直す：

$$I(x, \omega_o) = \int_{\Omega_i} L(\omega_i) T(x, \omega_i, \omega_o) d\omega_i, \quad (2)$$

$$T(x, \omega_i, \omega_o) \stackrel{\text{def}}{=} R(x, \omega_i, \omega_o) V(x, \omega_i) \max(0, \omega_i \cdot n), \quad (3)$$

視線が固定されると、 $T_x(\omega_i) = T(x, \omega_i, \omega_o)$ 、 $I_x = I(x, \omega_o)$ とできる。ここで x は対応する頂点を表現している。光の伝達関数を $T_x(\omega_i)$ とし、無限遠の照明を $L(\omega_i)$ とする。伝達関数を作る SRBF の中心を $\Xi T = \{\xi_{T,1}, \dots, \xi_{T,m}\}$ とし、光源を作る SRBF の中心を $\Xi L = \{\xi_{L,1}, \dots, \xi_{L,n}\}$ とする。それぞれ、単位球 S^2 上の点である。対応する bandwidth パラメータを $\Lambda T = \{\lambda_{T,1}, \dots, \lambda_{T,m}\}$ と $\Lambda L = \{\lambda_{L,1}, \dots, \lambda_{L,n}\}$ と定義する。これらは実数である。すると、 T_x と L は SRBF 関数 G と係数を用いて次のように表現できる。

$$T_x(\omega_i) \approx \sum_{i=1}^{n_i} t_{x,i} G(\omega_i, \xi_{T,i}; \lambda_{T,i}), \quad (4)$$

$$L(\omega_i) \approx \sum_{j=1}^{n_l} l_j G(\omega_i, \xi_{L,j}; \lambda_{L,j}), \quad (5)$$

ここで $G(\cdot, \xi; \lambda)$ は ξ を中心に、 λ を bandwidth に持つ SRBF である。以降、 $G(\omega_i, \xi_{T,i}; \lambda_{T,i})$ を T -SRBF、 $G(\omega_i, \xi_{L,i}; \lambda_{L,i})$ を L -SRBF と呼ぶことにする。式(2-5)を考えると、次のような式が得られる。

$$I_x = \mathbf{T}_x^T \mathbf{A} \mathbf{L}, \quad (6)$$

ここで $\mathbf{T}_x = [t_{x,1} \ t_{x,2} \ \dots \ t_{x,m}]$ は伝達係数、 $\mathbf{L} = [l_1 \ l_2 \ \dots \ l_n]$ は光源係数と呼ぶ。また行列 \mathbf{A} の m 行 n 列成分は、

$$A_{mn} = \int_{\Omega_i} G(\omega_i, \xi_{T,m}; \lambda_{T,m}) G(\omega_i, \xi_{L,n}; \lambda_{L,n}) d\omega_i. \quad (7)$$

となる。式(7)は spherical singular integral と呼ばれ、Gaussian SRBF を用いて効率よく計算される[18]。

4.2. 環境マップの推定

我々は環境マップを一様に分布した L-SRBF によって初期化する。L-SRBF の初期位置は正 20 面体を数回再分割することによって得られる。我々は 2 回の再分割を用い、162 個の L-SRBF を得た。この章では、ディフューズブラシ、スペキュラーブラシによる情報から L-SRBF の係数を推定する手法について述べる。

4.2.1. 基本の定式化

$\mathbf{T}^T \mathbf{x} = (\mathbf{T}^T \mathbf{A})^T$ とすると、式(6)は、

$$l_x = \mathbf{T}_x^T \mathbf{L}. \quad (8)$$

となる。ユーザがペイントした色を $I = \{I^1, \dots, I^{np}\}$ とし、塗られた頂点を $P = \{p_1, \dots, p_{np}\}$ とすると、システムは、レンダリングされた頂点色がユーザによって指定された色となるべく近くなるように、新しい環境マップを推定する。この推定問題の簡単に定式化すると、次のような最小二乗法が得られる。

$$\hat{\mathbf{L}} = \underset{\mathbf{L}}{\operatorname{argmin}} \sum_{i=1}^{n_p} (I_i^T - \mathbf{T}_{p_i}^T \mathbf{L})^2, \quad (9)$$

Under constrained な条件のもとで、ユニークな解を得るために、我々は regularization 項を追加する。式 9 は次のように変わる。

$$\hat{\mathbf{L}} = \underset{\mathbf{L}}{\operatorname{argmin}} \sum_{i=1}^{n_p} (I_i^T - \mathbf{T}_{p_i}^T \mathbf{L})^2 + \kappa \mathbf{L}^2, \quad (10)$$

ここで、 κ は regularization 因子である。我々は現在 $\kappa = 1.0e-4$ を使用している。

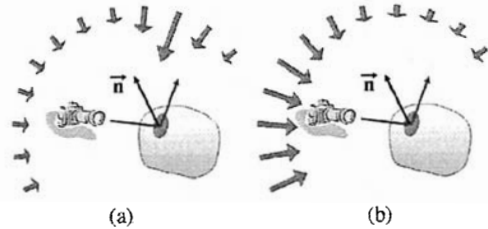


図 6 : ユーザはオレンジ色をサーフェスに塗っているとする。(a)スペキュラーブラシは鏡面反射方向に位置している光源により影響を及ぼす。この方向に位置する光源は、塗られた場所でのスペキュラー成分に寄与する可能性が高いためである。(b)ディフューズブラシは鏡面反射方向から離れた位置にある光源により影響を及ぼし、結果として視線方向に依存しにくいディフューズ成分を作り出す。

4.2.2. ディフューズ、スペキュラーブラシ

次に我々は上の推定アルゴリズムにディフューズブラシとスペキュラーブラシを組み込む。図 6 はそれぞれのブラシに対する我々のデザイン方法を示している。スペキュラーブラシが使用されると、鏡面反射方向に位置している光源はよりたくさん修正される (図 6(a))

が、一方でディフューズブラシが使用されると、鏡面反射方向から離れた位置にある光源がよりたくさん修正される (図 6(b))。

これらの特徴を実装するために、我々は重みベクトル $\mathbf{W} = [w_1 w_2 \dots w_{n_i}]^T$ を定義し、式(10)を次のように書き換える

$$\hat{\mathbf{L}} = \underset{\mathbf{L}}{\operatorname{argmin}} \sum_{i=1}^{n_p} (I_i^T - \mathbf{T}_{p_i}^T \mathbf{L})^2 + \kappa (\operatorname{diag}(\mathbf{W}) \mathbf{L})^2. \quad (11)$$

新しい重みベクトルはどの光源がよりたくさん変更されるかをコントロールする。つまり、 w_i が小さくなれば、 I_i はよりたくさん変化できるし、またその逆が言える。

\mathbf{W} を決めるためには、伝達係数ベクトルである $\mathbf{T}^T \mathbf{x}$ を使用する。この伝達係数は、どの光源が l_x を決めるのに大切か、ということを説明している (式(8))。我々は塗られた頂点 x ごとに、重みベクトルを次のように計算する。もし x がディフューズブラシで塗られていれば、重みは $w_{x,i} \propto I^1 / \max\{I^j\}$ で与えられ、もし x がスペキュラーブラシで塗られていれば、重みは $w_{x,i} \propto 1 - I^1 / \max\{I^j\}$ で与えられる。サーフェスから見えない (裏側にある) 光源の重みはゼロとする。最後に \mathbf{W} は全ての塗られた頂点に対し、 $\mathbf{W} = \Sigma \mathbf{W}_{p_i}$ と計算される。

4.3. 環境マップの回転

環境マップは L-SRBF の \mathbf{L} を回転させることで容易に回転できる。ユーザがドラッグ操作を行うと、システムは自動的に回転角度を計算し、マウスの下の色をキープしようとする。より正確には、計算処理は次のようになる (図 7)。マウスカーソルが p_0 から p_1 まで、スクリーン上を移動するとし、対応する 3 次元座標を v_0 と v_1 とする。システムは新しい光源の方向を最初の v_0 における照明効果 I_{v_0} と新しい v_1 における照明効果 I_{v_1} の差を最小にするように推定する。

我々は標準的な最小化問題を解く。回転行列を \mathbf{R} とすると、 \mathbf{R} は 3 つの回転角度 α, β, γ で表現される。システムは $I_{v_1} = I_{v_0}$ を満たすような最小の回転角度 ($d\alpha, d\beta, d\gamma$) を計算する。この問題は Lagrange multiplier で、

$$\min \{d\alpha^2 + d\beta^2 + d\gamma^2\} \quad \text{s.t.} \quad dI_{v_1} = \frac{\partial I_{v_1}}{\partial \mathbf{J}} d\mathbf{J},$$

と書ける。ここで $\mathbf{J} = [\alpha \ \beta \ \gamma]^T$ である。 dI_{v_1} は目的となる値 $I_{v_0} - I_{v_1}$ であり、システムは継続的にこの式を、バックグラウンドプロセスで解き、 \mathbf{J} を更新し続ける。この時、 $d\mathbf{J}$ には $\epsilon = 0.05$ が掛け算され、 \mathbf{J} に足される。

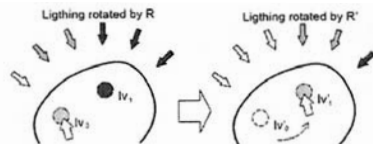


図 7 : ドラッグ操作による光源の回転。システムは v_1 と v_0 の照明効果を同一にするように \mathbf{R} を計算する。

5. 結果

本文中に掲載した3次元モデルは、全て測定および合成により生成されたBRDFを用いている。BRDFと推定アルゴリズムの性能、安定性は無関係であるが、BRDFがスペキュラーになればなるほど、ユーザは多くの入力が必要になる。

システムは照明を数秒で推定することができた(162個のL-SRBFあたり1~5秒)。推定に掛かる時間は3次元モデルの大きさには無関係であり、L-SRBFの数に依存する。

5.1. 3次元モデルの実写画像への挿入

3次元モデルを実写画像上に合成する際に、両者の照明効果の一貫性を保つ必要があるが、その際に本システムが便利である。画像を読み込んで、3次元モデルの背景として表示し、3次元モデルが背景になじむように、3次元モデル上にペイントしながらライトをデザインすることができる。図10に、これらの処理を行った合成写真の結果を示す。全ての3次元モデルは測定によって得られたBRDFを使用している。

5.2. ユーザテスト

提案手法の効率を評価するために、我々は現在、環境マップの編集によく使用されている組み合わせである、Adobe PhotoshopとHDRShopプラグインとの比較を行った。8人の情報科学を専攻する学生を被験者とした。被験者は3次元グラフィックス・ユーザインタフェースおよび提案手法の初心者である。まず各々の被験者はターゲット画像(図8(a))と、デフォルトの照明下に置かれたバニーモデル(図8(b))を与えられる。各々のシーンともに測定によって得られたアルミニウム・ブロンズBRDFを使用している。短時間のチュートリアルをしたあと、被験者はバニーモデルの照明を、ターゲット画像の照明に合わせるように、デザインすることを要求される。このとき提案システムかPhotoshop+HDRShopを使用する。両方のデザインツールで、被験者はターゲット画像内の色を取り、その色を照明のデザインに使用することを許されている。ソフトウェアの実験順序は被験者によって異なる。4人は提案手法を先に使用し、別の4人がPhotoshopを先に使用した。被験者は結果に満足がいくまで作業をすることができるが、時間制限を20分とした。ほとんどの被験者は、チュートリアルも含めて、ほぼ70分の時間を実験に費やした。図8に結果画像をいくつか示す。

図9(a)は被験者が各々のソフトウェアで費やした時間を示している。提案手法を使った被験者は、だいたいPhotoshopの75%の時間で作業を終えている。図9(b)は結果画像に対する主観評価を示している。16人が結果画像に投票を行った。このときの投票の基準は、どのくらい結果の画像がground truthの環境マップでレンダリングされた結果画像(図8(c))に似ているか、というものである。数字は各々のソフトウェアに対する投票数を示している。このスコアの二項検定を行った結果、提案手法とPhotoshopの効率の違いが有意に示された。これらの結果から、我々の手法は、より短い時間でPhotoshopよりもよりよい結果を出しているということが言える。

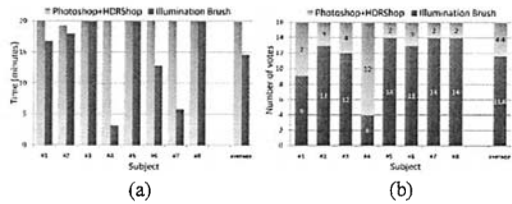


図9: デザインされたシーンの主観評価

参考文献

- [1] K. Anjyo and K. Hiramitsu. Stylized highlights for cartoon rendering and animation. *IEEE Comput. Graph. Appl.*, 23(4):54-61, 2003.
- [2] H. Biermann, I. Martin, F. Bernardini, and D. Zorin. Cut-and-paste editing of multiresolution surfaces. In *SIGGRAPH '02*, pages 312-321. ACM Press., 2002.
- [3] P. Debevec. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *SIGGRAPH*, pages 189-198, 1998.
- [4] S. Gumbold. Maximum entropy light source placement. In *VIS '02: Proceedings of the conference on Visualization '02*, pages 275-282, Washington, DC, USA, 2002. IEEE Computer Society.
- [5] T. Igarashi and J. F. Hughes. Clothing manipulation. In *UIST '02*, pages 91-100. ACM Press., 2002.
- [6] T. Jung, M. D. Gross, and E. Y.-L. Do. Light pen - sketching light in 3d. In *Proc. of CAAD Futures*, pages 327-338, 2003.
- [7] J. T. Kajiya. The rendering equation. In *SIGGRAPH*, pages 143-150, New York, NY, USA, 1986. ACM Press.
- [8] J. K. Kawai, J. S. Painter, and M. F. Cohen. Radiosity optimization: goal based rendering. In *SIGGRAPH*, pages 147-154, 1993.
- [9] Y. Li, S. Lin, H. Lu, and H.-Y. Shum. Multiple-cue illumination estimation in textured scenes. In *Int'l Conf. on Computer Vision*, pages 1366-1373, Nice, France, 2003.
- [10] K. Nishino and S. K. Nayar. Eyes for relighting. *ACM Trans. Graph. (SIGGRAPH)*, 23(3):704-711, 2004.
- [11] F. Pellacini, P. Tole, and D. P. Greenberg. A user interface for interactive cinematic shadow design. In *SIGGRAPH*, pages 563-566, 2002.
- [12] P. Poulin, K. Ratib, and M. Jacques. Sketching shadows and highlights to position lights. In *Proc. of Conference on Computer Graphics International*, pages 56-63, 1997.
- [13] R. Ramamoorthi and P. Hanrahan. A signal-processing framework for inverse rendering. In *SIGGRAPH*, pages 117-128, 2001.
- [14] I. Sato, Y. Sato, and K. Ikeuchi. Illumination from shadows. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(3):290-300, 2003.
- [15] C. Schoeneman, J. Dorsey, B. Smits, J. Arvo, and D. Greenburg. Painting with light. In *SIGGRAPH*, pages 143-146, 1993.
- [16] R. Shaded and D. Lischinski. Automatic lighting design using a perceptual quality metric. *Computer Graphics Forum*, 20(3):1067-7055, 2001.
- [17] A. Shesh and B. Chen. Crayon lighting: Sketch-based

illumination of models. Proceedings of Pacific Graphics 2006, pages 113–116, 2006.

[18] Y.-T. Tsai and Z.-C. Shi. All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. ACM Trans.

Graph.(SIGGRAPH), 25(3):967–976, 2006.

[19] Y. Wang and D. Samaras. Estimation of multiple directional light sources for synthesis of augmented reality images. Graph. Models, 65(4):185–205, 2003.

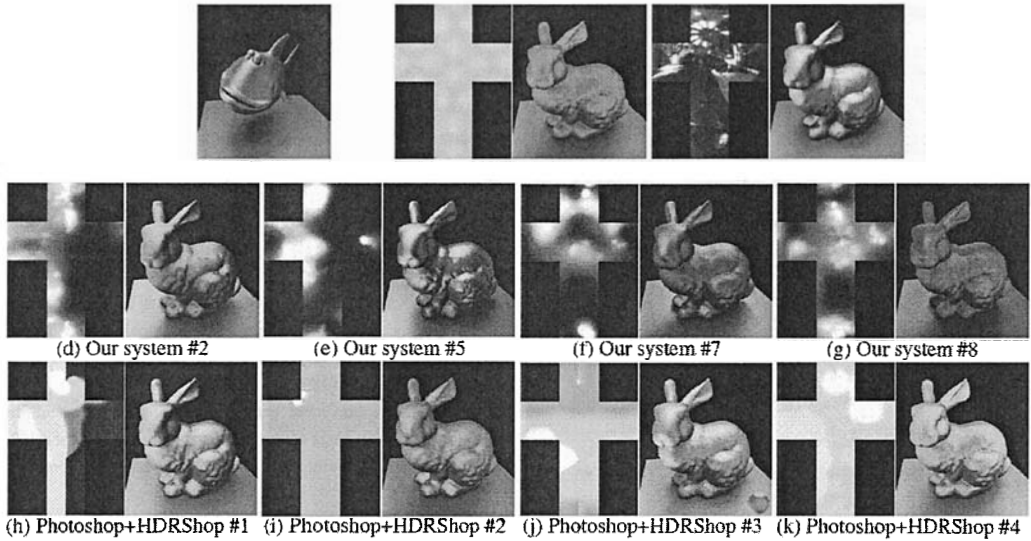


図 8 : ユーザテストの結果。提案手法と Photoshop+HDRShop の組み合わせを比較した。

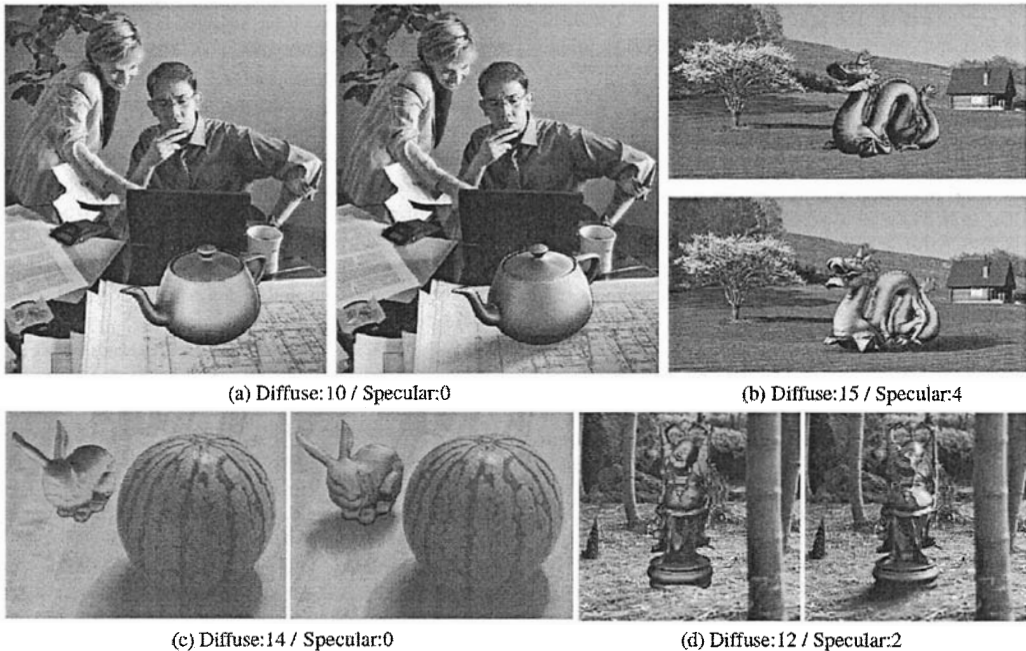


図 10 : 3次元モデルを実写画像上に合成した例。t-pot、ドラゴン、バニー、ブッダの各モデルはそれぞれ、計測によって得られた、テフロン、ゴールドメタリックペイント、ピンクプラスチック、そして白色な光沢のある物質のBRDFを持つ。図の下に示した数字は、証明デザインに使用した各ブラシの操作回数である。