

# Kinect を用いたリライティング

## Relighting with Kinect

山田 裕貴<sup>†</sup>, 金森 由博<sup>†</sup>, 福井 幸男<sup>†</sup>, 三谷 純<sup>†, ††</sup>

Hiroki Yamada<sup>†</sup>, Yoshihiro Kanamori<sup>†</sup>, Yukio Fukui<sup>†</sup> and Jun Mitani<sup>†, ††</sup>

**Abstract** Relighting has been used in many image productions. Relighting requires object shape information and reflectance, and previously one had to manually retouch the image or use special equipment. We therefore present a novel system for relighting with Kinect, which is a relatively inexpensive depth camera. First, we capture target objects and calculate their shapes. We then estimate reflectance from illumination that is calculated from an environment map taken at the same place. Finally, we relight the object using new illumination. The direction and intensity of the light can be adjusted by the user. This system enables the user to relight objects without complicated operations or expensive equipment.

キーワード: 画像合成, リライティング, 反射率推定, Kinect

### 1. ま え が き

物体の画像を背景と合成する画像編集技術は、映像制作などにおいてたびたび用いられる技術である。画像合成は、単純に元の画像を張り付けただけでは、撮影時の照明の違いから違和感のある結果になることが多い。そのため、自然な合成画像を得るために物体の画像の色を調整する手法が研究されてきた。

光の物理的挙動を計算して物体の画像の色を変える方法は、リライティング (Relighting) と呼ばれる。リライティングにより合成先の背景画像での照明によって物体を「照らしなおす」ことで、物体の画像の色を現実にその画像の中にあるような自然な見た目に調整することができる。

リライティングを行うためには、物体の形状・反射率といった情報が必要となるが、1枚の画像のみからこれらの情報を抽出することは難しく、既存手法ではユーザ入力の手間や金銭的コストがかかってしまい、広い用途で使用できないという問題がある。

そこで本研究では、デプス情報を取得できるカメラの一つである Microsoft 社の Kinect を使用した、新しいリライティングのシステムを提案する。Kinect は比較的安い価格で市販されており入手が容易であるという利点がある。本

研究では Kinect で取得できるデプス情報を利用して、ユーザ入力を行わずに物体の形状を得る。さらに、物体の形状情報を利用して反射率を推定しリライティングを行う。本研究では、デプス情報のノイズに対して RGB 画像の情報を利用した平滑化を行う。リライティングの結果はユーザの対話的な操作で、望ましい結果になるように調整することができる。

### 2. 関 連 研 究

照明効果を計算してリライティングを行うためには、物体の形・反射率を得る必要がある。Okabe ら<sup>1)</sup>は物体の形状を直感的なユーザ入力により補う方法を提案した。ユーザ入力により、特殊な撮影装置を使わずに物体形状を得ることができるが、ユーザ入力には手間がかかる。これに対して本研究では、Kinect で取得したデプス情報を使用して物体の形状を求める。これは特別なユーザ入力が必要としない。Chabert ら<sup>2)</sup>は高速に変化する照明装置とハイスピードカメラを使い、さまざまな光源で照らされた画像セットを撮影して、それらから人物の形状と反射率を測定した。この方法は、より正確なリライティングを実現しているが、専用の大掛かりな撮影装置を用意する必要がある。これに対し、本研究で撮影に使用する Kinect は比較的安価で簡単に用意することができ、持ち運びも容易である。

### 3. 提 案 手 法

#### (a) リライティング

本研究では非発光物体に限定し、遮蔽を無視する。レンダリング方程式<sup>3)</sup>からそれらの項を除くと、物体の色は次の式で表される。

2012年3月30日受付, 2012年10月29日再受付, 2013年3月11日採録

<sup>†</sup>筑波大学

(〒 305-8573 茨城県つくば市天王台 1-1-1, TEL 029-853-6246)

<sup>††</sup>JST ERATO

(〒 112-0002 東京都文京区小石川 1-28-1, TEL 029-853-2333)

<sup>†</sup>The University of Tsukuba

(1-1-1, Tennodai, Tsukuba-shi, Ibaraki, 305-8573, Japan)

<sup>††</sup>JST ERATO

(1-28-1, Koishikawa, Bunkyo-ku, Tokyo, 112-0002, Japan)

$$L(\mathbf{x}, \omega) = \int_{\Omega} f_r(\mathbf{x}, \omega', \omega) L_i(\mathbf{x}, \omega') (\omega' \cdot \mathbf{n}) d\omega' \quad (1)$$

ここで、 $L(\mathbf{x}, \omega)$  は物体の色を表す。 $\mathbf{x}$  は物体上のある点で、 $\omega$  は点  $\mathbf{x}$  から視点方向に向かう単位ベクトルを表している。 $\Omega$  は点  $\mathbf{x}$  における物体の法線  $\mathbf{n}$  の方向を向いた上向き単位半球面であり、 $\omega'$  は光の入射する方向を表している。 $\mathbf{n}$  および  $\omega'$  は単位ベクトルである。 $L_i(\mathbf{x}, \omega')$  は入射光を表す。 $f_r(\mathbf{x}, \omega', \omega)$  は BRDF(Bidirectional Reflectance Distribution Function) である。

物体の反射モデルは拡散反射と仮定することで BRDF は

$$f_{r,d}(\mathbf{x}) = \frac{\rho_d(\mathbf{x})}{\pi} \quad (2)$$

となる<sup>4)</sup>。ここで  $\rho_d$  は拡散反射率を表す。視点方向、入射方向に依存しない関数になるため積分の中から取り出すことができ、式 (1) は次のようになる。

$$L(\mathbf{x}) = \frac{\rho_d(\mathbf{x})}{\pi} \int_{\Omega} L_i(\mathbf{x}, \omega') (\omega' \cdot \mathbf{n}) d\omega' \quad (3)$$

簡単化のために BRDF の項と積分の項を

$$\frac{\rho_d(\mathbf{x})}{\pi} = \rho'_d(\mathbf{x}) \int_{\Omega} L_i(\mathbf{x}, \omega') (\omega' \cdot \mathbf{n}) d\omega' = E(\mathbf{x}) \quad (4)$$

とおくと式 (3) は

$$L(\mathbf{x}) = \rho'_d(\mathbf{x}) E(\mathbf{x}) \quad (5)$$

となる。この式から物体の色は反射率  $\rho'_d(\mathbf{x})$  と放射照度  $E(\mathbf{x})$  の積で計算できることがわかる、反射率は次の式で求まる。

$$\rho'_d(\mathbf{x}) = \frac{L(\mathbf{x})}{E(\mathbf{x})} \quad (6)$$

新しい照明でリライティングした物体の色  $L'(\mathbf{x})$  は、新しい照明の放射照度  $E'(\mathbf{x})$  をかけ、次の式で求められる。

$$L'(\mathbf{x}) = \rho'_d(\mathbf{x}) E'(\mathbf{x}) \quad (7)$$

#### (b) 物体の色の取得

Kinect は  $640 \times 480$  の解像度で RGB 画像情報を取得できる。ノイズを低減するためにガウシアンフィルタを用いて  $3 \times 3$  のカーネルサイズで平滑化を行う。

#### (c) 法線の計算

Kinect で取得したデプス情報のノイズを低減するためにジョイントバイラテラルフィルタ<sup>9)</sup>を使用して、RGB 画像情報を利用した平滑化を行う。ジョイントバイラテラルフィルタでは、RGB 画像の勾配が大きいピクセルではデプスを弱く平滑化し、RGB 画像の勾配が小さいピクセルでは強く平滑化する。これにより、Kinect で測定できない細かい凹凸の情報を RGB 画像の勾配から得ることで、デプスに反映することができる。しかし、図 2 左の人物画像のような、はっきりしたテクスチャのある物体では、本来存在しない

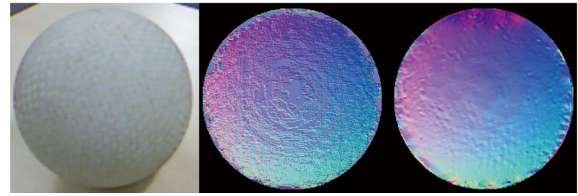


図 1 左：対象とする物体（ソフトバレーボール）の RGB 画像。中：平滑化前のデプスで計算した法線マップ。右：平滑化後のデプスで計算した法線マップ。平滑化によりなめらかな法線が得られている。 Left: A RGB image of the target object (soft volleyball). Center: A normal map of the raw depth map. Right: A normal map of the filtered depth map. The normal map is smoothed by filtering.

凹凸がデプスに現れてしまうという問題がある。その場合は、ガウシアンフィルタをデプスに適用して平滑化を行う方が結果がよくなる場合がある。図 2 の結果では、左の画像ではガウシアンフィルタを、右の画像ではジョイントバイラテラルフィルタを使用した。

図 1 に計算した法線マップの例を示す。法線マップは法線の  $x, y, z$  成分を R,G,B 色情報で描画したものである。法線は次の式で計算される。

$$\mathbf{n} = \frac{[\mathbf{X}_{u+1,v} - \mathbf{X}_{u,v}] \times [\mathbf{X}_{u,v+1} - \mathbf{X}_{u,v}]}{\|[\mathbf{X}_{u+1,v} - \mathbf{X}_{u,v}] \times [\mathbf{X}_{u,v+1} - \mathbf{X}_{u,v}]\|} \quad (8)$$

ここで、 $\mathbf{X}_{u,v}$  はピクセル座標  $(u, v)$  における物体表面上の点の視点座標系での点  $(x, y, z)$  を表す。ピクセル座標から視点座標系への変換は Kinect の透視投影の逆変換で求まる。

#### (d) 照度の計算

照明には環境マップを光源として使う。合成する物体の画像を撮影する際に、鏡面球を置いて環境マップを撮影する。合成先の環境マップは、鏡面球を撮影できない場合は、背景画像を緯度・経度方向の  $180^\circ \times 360^\circ$  を長方形に展開した環境マップとみなして計算する。これは Khan ら<sup>9)</sup>の方法を参考にしたもので、擬似的な環境マップによるリライティングでもそれらしい結果を得ることができる。放射照度は Ramamoorthi ら<sup>7)</sup>の提案した手法で計算する。これは、球面調関数の 2 次までの項の九つの係数を用いて、放射照度を近似的に表す方法で、これによりレンダリングの高速化を図る。

## 4. 結果

本システムは C++ 言語により実装した。実行環境は Microsoft Windows 7 Professional, Intel Core i 7-2600 CPU(3.40Ghz), メインメモリー 4GB, GPU に NVIDIA GeForce GTX 460 を搭載した PC である。放射照度の計算には  $512 \times 512$  ピクセルの環境マップでおよそ 40 ミリ秒、ジョイントバイラテラルフィルタを使用したデプス平滑化はカーネルサイズ 21 で  $640 \times 480$  のデプスマップに対しておよそ 3.7 秒かかる。放射照度と平滑化したデプスマップを使用してリライティングを行う。リライティングはおよそ 16 ミリ秒で計算できる。

本システムによってリライティングを行い、背景画像に

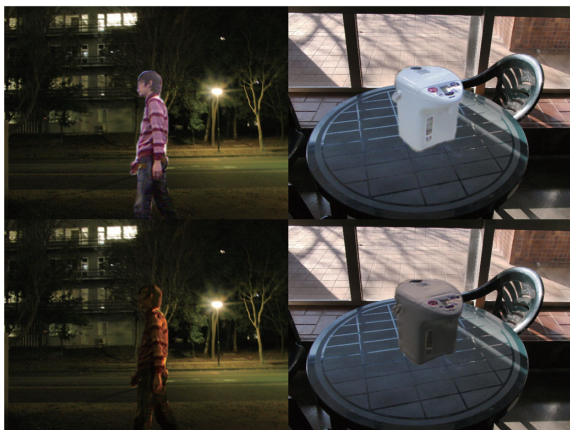


図2 人物(左)と湯沸かしポット(右)の画像を背景写真に合成した画像。上段:リライティングを行っていない合成画像。下段:本システムによってリライティングを行った合成画像。リライティングによりそれらしい合成画像を得ることができる。  
A man (left) and a kettle (right) are composed to background images. Upper row: Composite pictures without relighting. Bottom row: with relighting by our system. Relighting makes pictures more plausible.

合成した結果を図2に示す。リライティングによって違和感の少ない合成結果が得られたことがわかる。Color Transferを用いた方法では、対応できない強い光源のある画像でも、物体の形状と光源の位置を考慮して色が調整されている。

ユーザは、よりそれらしい結果を得るために、照明の方向をドラッグ操作で変更したり、照明の強さの倍率を調整することができる。また、本システムでは、リライティング結果とリライティング前の元画像の色を線形混合することで、より良い結果になる場合があることが実験的にわかったので、この比率を調整することができるようにした。これらの操作は対話的に行うことができる。

### 5. 制約

本研究で扱う物体は、拡散反射と仮定しているの、周囲の景色が映り込むような光沢のある物体を合成すると、映り込みの不一致により不自然な結果になる場合がある。本システムでは、物体が背景に落とす影や、物体に落ちる影を計算できない。このような背景と物体の相互的な影響を扱うには、背景のジオメトリ情報が必要になる。背景のジオメトリ情報を得る方法としては、Karschら<sup>9)</sup>が行ったように1枚の画像からユーザ入力や最適化計算によって求める方法や、Newcombeら<sup>9)</sup>のようにKinectを使ってスキャンすることで取得するといった方法などが考えられる。物体と背景が互いに影響する効果を計算することができれば、より真実味のある合成結果を得ることができると思われる。

### 6. むすび

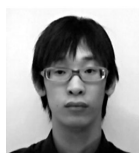
本研究は、Kinectを使いリライティングを行うシステムを提案した。本手法では、既存手法のような複雑なユーザ入力や特別な撮影装置を必要とせずに物体形状を求めることができる。物体形状の情報と環境マップを利用して、物体

にあたる光を計算することで物体の反射率を推定した。デプス情報のノイズを低減するために、RGB画像情報を利用したジョイントバイラテラルフィルタによるデプスの平滑化を行った。リライティングの結果は実時間で計算される。

今後の研究として、ノイズの多いデプス情報からより正確な形状を復元する手法が期待される。また、計算時間の向上により、撮影からリライティングまでを実時間で処理できるようになれば、より実用的なシステムになると思われる。

### 〔文 献〕

- 1) M. Okabe, G. Zeng, Y. Matsushita, T. Igarashi, L. Quan, and H.-Y. Shum. "Single-view relighting with normal map painting". In *Proceedings of Pacific Graphics 2006*, pp. 27–34 (2006)
- 2) C.-F. Chabert, P. Einarsson, A. Jones, B. Lamond, W.-C. Ma, S. Sylvan, T. Hawkins, and P. Debevec. "Relighting human locomotion with flowed reflectance fields". In *ACM SIGGRAPH 2006 Sketches*, SIGGRAPH '06, pp. 76. New York, NY, USA (2006)
- 3) J. T. Kajiya. "The rendering equation". In D. C. Evans and R. J. Athay, editors, *SIGGRAPH '86 Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pp. 143–150, ACM New York, NY, USA (1986)
- 4) N. Kurachi. *The Magic of Computer Graphics*. A K Peters/CRC Press (June. 2011)
- 5) G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama. "Digital photography with flash and no-flash image pairs". *ACM Trans. Graph.*, Vol. 23, No. 3 pp. 664–672 (Aug. 2004)
- 6) E. A. Khan, E. Reinhard, R. W. Fleming, and H. H. Bulthoff. "Image-Based material editing". In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pp. 654–663, New York, NY, USA (2006)
- 7) R. Ramamoorthi and P. Hanrahan. "An efficient representation for irradiance environment maps". In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pp. 497–500, New York, NY, USA (2001)
- 8) K. Karsch, V. Hedau, D. Forsyth, and D. Hoiem. "Rendering synthetic objects into legacy photographs". In *Proceedings of the 2011 SIGGRAPH Asia Conference*, SA '11, pp. 157:1–157:12, New York, NY, USA (2011)
- 9) R. Newcombe, S. Izadi, O. Hilliges, D. Molyneux, D. Kim, A. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. "KinectFusion: Real-time dense surface mapping and tracking". In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pp. 127–136 (Oct. 2011)



やまだ ひろき  
山田 裕貴 2012年、筑波大学情報学群情報科学類卒業。現在、筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻博士前期課程在学中。



かなもり よしひろ  
金森 由博 2009年、東京大学情報理工学系研究科博士課程修了。博士(情報理工学)。同年、筑波大学に勤務し、現職は筑波大学システム情報系・助教。リアルタイムレンダリングやビジュアルシミュレーションに興味を持つ。最近は特に、現実世界の現象を再現できる画像編集技術に取り組んでいる。



ふくい ゆきお  
福井 幸男 1973年、京都大学工学部卒業、同年、(株)日立製作所入社。1980年、東京大学院修士課程修了、同年、通産省工業技術院製品科学研究所入所。同院生命工学工業技術研究所を経て、1998年筑波大学に転任、現在は筑波大学システム情報系教授。3次元形状の生成と設計応用、感覚操作特性に関する研究等に従事。



みやた じゅん  
三谷 純 2004年に東京大学工学部卒業、同大学院修了、博士(工学)。理化学研究所研究員を経て、2006年より、筑波大学に勤務し、現職は筑波大学システム情報系准教授。コンピュータグラフィックスにおける形状モデリングの研究に従事。曲面を持つ立体折紙を設計する技法に関する研究などを行っている。