Vol.2014-CVIM-190 No.36
2014/1/24

# RGB-D カメラを用いた動的光源環境へのリライティング

池田 拓也[*1]　フランソワ ドゥ ソルビエ[*1]　斎藤 英雄[*1]

## Relighting with Dynamic Light Environments Using an RGB-D Camera

Takuya Ikeda[*1]　Francois de Sorbier[*1]　and　Saito Hideo[*1]

**Abstract** – リライティングとは，ある光源環境で撮影された対象に対して，任意の光源環境に合わせてその陰影を変更する技術である．リライティングを行うには，対象物体の法線分布，反射特性，光源環境が必要である．本稿では，RGB-D カメラを用いた実時間リライティング手法を提案する．本手法では，正確な物体領域抽出のためにノイズを含む距離画像の修正を行う．領域抽出を行った後，対象物体の法線分布を推定する．光源環境は，観測された鏡面球から取得し，任意の環境に対してリライティングを行う．また，実時間処理を実現するために，GPU を用いた並列処理を行った．我々の実装では，約 10fps の速度でリライティングが可能である．本実験では，提案手法の有効性を確認するために，対象物体に対して事前に撮影した動的光源環境へリライティングを行い，提案手法の有効性を示した．

**Keywords** : Relighting, Lambertian model, RGB-D camera, GPU

## 1　はじめに

ある照明条件の元で撮影された人物や物体などに対し，他の照明条件に合わせて陰影を変更する処理を，リライティング (Relighting) と呼ぶ．この技術は，主に映画や放送などの映像制作で利用されている．映画制作では，CG で合成されたシーンに合わせて演者の顔や服の陰影を変更することで，光学的整合性の取れたリアルな映像を生成している [1]．また，現実空間に仮想環境から情報や仮想物体を重畳して表示する拡張現実感（Augmented Reality，以下 AR）においても，リライティングは重要である．仮想物体の反射特性やシーンの光源情報を用いて現実空間との光学的整合性が取れた AR を実現できる [2]．さらに，商用アプリケーションへの応用が考えられる．例えば，衣服の試着を行うシーンでの利用である．ウェディングドレスやドレスなどを試着したユーザに対して，リライティングを行うことで，ユーザが結婚式場やパーティ会場にいるような仮想体験が可能である．このシステムは，ユーザに対して衣服購入の判断材料となりえる．映画や放送分野の映像制作ではオフラインで処理を行うのに対し，後者のアプリケーションでは，実時間で行うことが重要となる．なお，本論文で述べる実時間は，撮影された対象物体やシーンに対して，毎フレームリライティングを行う事とする．

リライティングを行うには，対象物体の 3 次元形状（法線分布），物体の反射特性，対象物体を撮影した環境（以下，Src 環境）とリライティングを行う環境（以下，Dst 環境）の光源情報が必要である．一般的には，画像中の物体領域を抽出し，上記の要素を用いて Dst 環境に合わせて陰影を変更，最後に Dst 環境の背景に合成して結果画像を得る．

従来研究として，Zhen ら [3] は，人物の顔を対象とし，1 枚の画像からリライティングを行う手法を提案している．この手法では，顔のモーフィングモデルを用いて対象となる顔の 3 次元形状を推定し，他のシーンの光源環境に合わせてリライティングを行っている．Zhen らの手法では，反射特性を拡散反射のみとし，Lambertian モデルを仮定している．また，撮影時の光源分布と目的のシーンの光源分布の比を計算することで拡散反射係数をキャンセルし，反射特性を求めずにリライティングを行う事が可能である．本論文で提案する手法では，高速化を優先するために，Zhen らと同様に物体の反射特性を拡散反射のみとし，Lambertian モデルを仮定する．

Debevec ら [4]，Wenger ら [5] は，Light Stage と呼ばれる大量の点光源を備えた装置を用いて，様々な方向から光源を照射した陰影画像を取得している．その後，取得した画像群を用いて Reflectance Functions と呼ばれる反射分布を作成し，任意の光源環境にリライティングする手法を提案している．この手法では，物体の 3 次元形状を得る必要が無く，反射分布を用いた高精度なリライティングが実現されている．Wenger らの手法では，高速撮影が可能なハイスピードカメラを用いることで，動的物体に対してリライティングが可能となっている．しかし，Light Stage を用いた手法では，それで撮影できないような環境に存在する物体に対してリライティングを行うことはできない．また，撮影された画像もしくは動画に対して，オフライ

[*1] 慶應義塾大学
[*1] Keio University

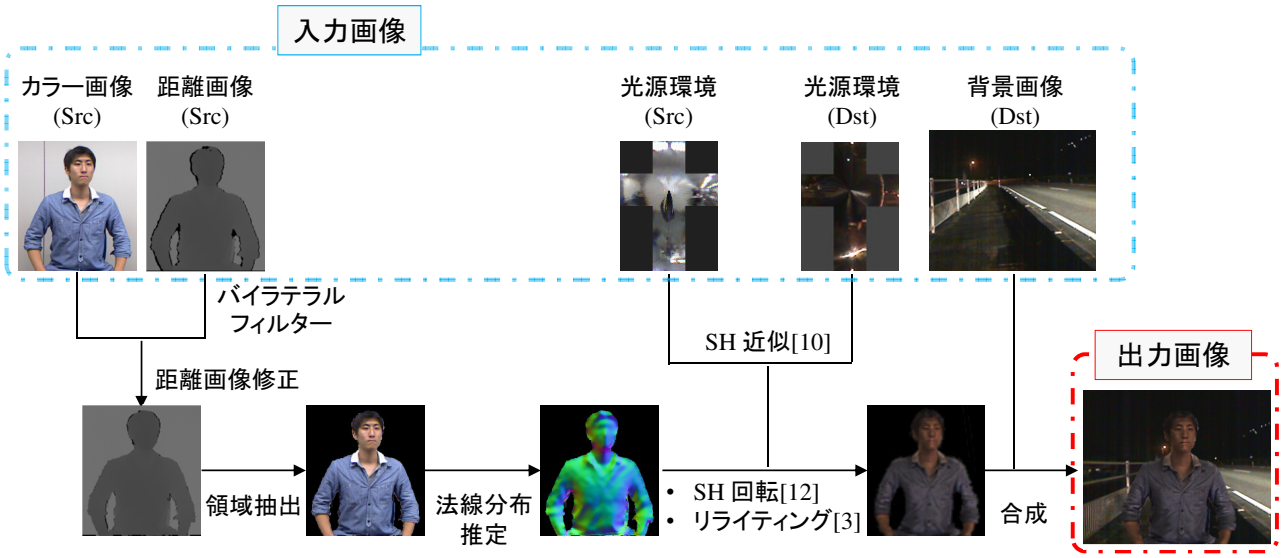



図 1　提案するリライティング手法の流れ.
Fig. 1　Proposed relighting flow.

ン処理でリライティングを行っており，実時間リライティングは実現できていない．

従来手法に対し，本論文では RGB-D カメラを用いた実時間リライティング手法を，新たに提案する．RGB-D カメラは，カラー画像と距離画像を実時間（約 30fps）で取得でき，実時間リライティングの実現に有効である．提案手法は，動的物体を対象とし，対象領域を任意の光源環境に合わせてリライティングし，他のシーンに合成するというものである．そのため，対象物体の正確な領域抽出が必要となる．また，リライティングの精度は，光源の方向を考慮するために用いる法線分布の精度に大きく依存する．以上の点から，本研究は対象の領域抽出，法線分布推定，実時間処理に着目し，それらに対して独自の工夫を施すことにより目的の実時間リライティングを可能にするものである．提案手法は，任意の環境において，動的物体や動的 Dst 環境に対してリライティングが可能であり，前述した AR への利用や商用アプリケーションへの応用に期待できる．

## 2　提案するリライティング手法

本章では，提案するリライティング手法について述べる．提案手法の流れを図 1 に示す．本システムでは，RGB-D カメラ 1 台と鏡面球 1 つを用いる．対象物体を撮影したカラー画像と距離画像，Src 環境と Dst 環境の光源分布，Dst 環境の背景画像を入力とする．なお，シーンの光源分布は観測された鏡面球より取得する．物体領域抽出は，距離値に閾値を設けて行う．この時，正確な領域抽出のために距離画像を修正する．次に，光源の考慮する範囲を決定するために，修正距離画像から法線分布を推定する．リライティング処理では，光源分布を SH に近似して用いる．Src 環境と Dst 環境の光源分布の比を計算することで陰影を変更した画像を取得し，Dst 背景画像に合成する．なお，後述する実験では，GPU を用いた処理の高速化を行う．

### 2.1　距離画像修正

対象物体を他のシーンに合成するために，対象の領域抽出を行うことが必要となる．提案手法では，距離画像に閾値を設けて領域抽出を行う．しかし，距離画像に含まれるノイズやカメラ間の視差により，距離画像とカラー画像で物体の境界が一致していないことが多い．そこで，境界を一致させるために，物体の境界周辺領域を抽出し，その領域の距離値を修正する．距離画像の修正は Chen ら [6] の手法をベースにしており，大きく分けて 2 つのステップからなる．1 つ目のステップは，対象物体と背景の境界周辺領域を検出し，その領域の距離値を空値とする．2 つ目のステップは，空値とした領域の距離値を，その近傍の距離値を用いて補間した値に置換する処理である．これにより，カラー画像と物体の境界が一致した距離画像を得る．

1 つ目のステップでは，まず入力距離画像に対して閾値処理を行い，物体領域を抽出する．その後，膨張処理により，物体領域を包含するマスク画像を得る (図 2 (a))．次に，物体の境界周辺領域を抽出するため，カラー画像と距離画像のマスク領域内においてそれぞれ，Canny のエッジ検出を行う．検出されたエッジに対して，$n \times n$ の正方領域を 1, その他を 0 とするマスク画像を作成する．カラー画像，距離画像のエッジに対して作成したマスク画像をそれぞれ $M_c$，$M_d$ とすると，物体の境界周辺領域を抽出したマスク画像 $M_f$ は，2 つのマスク画像の積，

$$M_f = M_d AND M_c \tag{1}$$

によって求める．カラー画像のエッジは物体境界だけでなく，テクスチャによるエッジも検出してしまうが，距離画像のエッジでは，物体境界のみ検出される．ここで，物体の境界周辺領域のために，カラー画像，距離画像のエッジに対する正方領域の 1 辺の長さをそれぞれ $n_c$，$n_d$ とすると，$n_c < n_d$ の条件を満たすように設定する．これにより，図 2(b) のような境界周辺領域を抽出する事ができる．

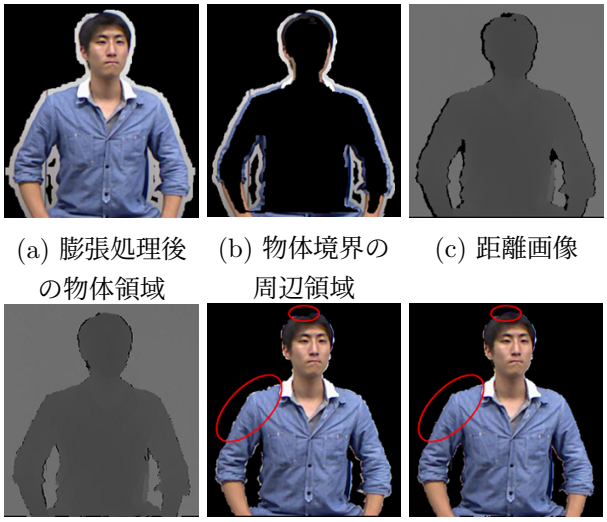
次に，2 つ目のステップでは，検出された領域の距離値を空値とし，それらを周辺の画素（空値とした画素を除く）の距離値から補間する．距離値の補間は，Chen ら [6] の手法と同様に、物体領域については物体領域内の距離値から補間し、背景領域については背景領域の距離値から補間する．Chen らは region growing を用いて，物体と背景を区別しているのに対して，本研究では k-means クラスタリングを用いたカラー画像の領域分割を行うことによって，物体と背景の領域を区別する．この点が，本論文と Chen らの手法の相違点であり，k-means クラスタリングを用いるのは，GPU で処理するアルゴリズム [7] を適用し，高速化するためである．

カラー画像の領域分割後，RGB-D カメラより取得した距離画像 $D$ に対して，

$$D^t(\boldsymbol{u}) = \frac{1}{k(\boldsymbol{u})} \sum_{\boldsymbol{v}\in\Omega_s, D^{t-1}\notin 0} g_s(\boldsymbol{u}-\boldsymbol{v}) g_c(i(\boldsymbol{u})-i(\boldsymbol{v})) D^{t-1}(\boldsymbol{v}) \tag{2}$$

を計算し，補間した距離画像を得る．ここで，$g_s$ は画素空間の重み，$g_c$ は色空間の重みをパラメータとして持つガウシアン関数であり，各標準偏差は $\sigma_s$, $\sigma_c$ で表される．$\Omega_s$ は，画素 $\boldsymbol{u}$ を中心とする，一辺 $n_s$ の長さの正方領域である．$\boldsymbol{u}-\boldsymbol{v}$ は画像空間，$i(\boldsymbol{u})-i(\boldsymbol{v})$ はカラー画像の色空間におけるユークリッド距離を表している．$D^{t-1}(\boldsymbol{v})$ は空値ではない近傍画素 $\boldsymbol{v}$ の距離値である．$k(\boldsymbol{u})$ は，$k(\boldsymbol{u}) = \sum_{\boldsymbol{v}\in\Omega_s, D^{t-1}\notin 0} g_s(\boldsymbol{u}-\boldsymbol{v}) g_c(i(\boldsymbol{u})-i(\boldsymbol{v}))$ によって算出される正規化要素である．距離値を補間する際は，k-means クラスタリングによって付けられたラベルと同じラベルの画素の距離値を用いる．つまり，画素 $\boldsymbol{u}$ の距離値は，$\boldsymbol{u}$ のラベルと同じラベルを持つ画素 $\boldsymbol{v}$ の距離値のみを用いる．これにより，前述した物体と背景の領域を区別して補間を行うことができる．この時，$\Omega_s$ の領域内には，空とした画素と距離値を持つ画素が存在する．本研究では，領域内に 2 画素以上距離値を持つ画素が存在する場合に，式 (2) を適用する．また，式 (2) を繰り返し $t$ 回行い，$D^t$ を毎回更新することで，最終的に空値とした画素が補間された修正距離画像 $D_m$ を得る．

図 2(c)，(d) はそれぞれ RGB-D カメラから取得した距離画像と修正後の距離画像，図 2(e)，(f) は各距離画像に対して物体領域を抽出した結果画像である．本手法により，図 2 の赤丸で示された人物の頭や衣服の境界に合わせて距離画像が修正され，領域抽出が改善されていることが分かる．



(a) 膨張処理後の物体領域　(b) 物体境界の周辺領域　(c) 距離画像

(d) 修正距離画像　(e) 抽出領域（距離画像）　(f) 抽出領域（修正距離画像）

図 2　距離画像の修正
Fig. 2　Depth map modification.

### 2.2　法線分布推定

次に，修正距離画像を用いて物体領域の法線分布を推定する．法線推定の前に，距離画像とカメラの内部パラメータ $K$ を用いて，$V(\boldsymbol{u}) = D_m K^{-1}[\boldsymbol{u}, 1]$ を計算し，画像形式で保持された 3 次元点群マップ $V$ を得る．カメラの内部パラメータについては，RGB-D カメラに関するライブラリー OpenNI [8] を用いた．まず，カメラから得られる距離画像は離散的な値を持っているため，滑らかな距離画像にするためにバイラテラルフィルタを用いたフィルタ処理を行う．修正距離画像 $D_m$ に対して，

$$D_b(\boldsymbol{u}) = \frac{1}{k'(\boldsymbol{u})} \sum_{\boldsymbol{v}\in\Omega_{s'}} g_{s'}(\boldsymbol{u}-\boldsymbol{v}) g_d(D_m(\boldsymbol{u})-D_m(\boldsymbol{v})) D_m(\boldsymbol{v}) \tag{3}$$

を算出し，ノイズを軽減した距離画像 $D_b$ を得る．ここで，$g_{s'}$ は画素空間の重み，$g_d$ は距離空間の重みをパラメータとして持つガウシアン関数，各標準偏差は $\sigma_{s'}$, $\sigma_d$ とする．$\Omega_{s'}$ は，画素 $\boldsymbol{u}$ を中心する，一辺 $n_{s'}$ の長さの正方領域である．$D_m(\boldsymbol{u}) - D_m(\boldsymbol{v})$ は距離値のユークリッド距離を表しており，$k'(\boldsymbol{u})$ は正規化要素である．

フィルター処理後，Stefan らの提案した手法 [9] を適用する．Stefan らは 3 次元点群マップに対して積分画像を計算し，法線を推定する手法を提案している．9 つの積分画像を用いて，分散共分散行列を構成し，そ

の固有ベクトルから法線方向を得る．積分画像を用いる事で，3 次元座標の平均を計算し，なめらかな法線分布を得ることができる．また，積分画像を用いる事で任意の矩形中の平均値算出が容易となり，処理時間が少ないという利点がある．

しかし，Stefan らが提案した 2 つの手法では，物体境界周辺の法線が算出できないという問題がある．これは，物体と背景の境界では，距離値が大きく異なっているため，誤った法線方向が推定されるからである．そこで，そのような領域では，着目画素の近傍画素とのベクトルを 2 つ算出し，それらの外積を法線ベクトルとする．具体的には着目画素 $\boldsymbol{u} = (u, v)$ の法線 $N(\boldsymbol{u})$ は，$N(\boldsymbol{u}) = (V(u+1, v) - V(u, v)) \times (V(u, v+1) - V(u, v))$ を計算後，$N(\boldsymbol{u})$ を正規化することによって得る．以上より，物体領域の法線分布を得る．

### 2.3 光源分布の取得

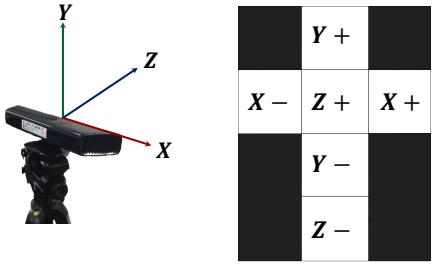
提案手法では，半径が既知の鏡面球を観測することで光源分布を取得する．鏡面球は，周囲の環境を反射するため，広範囲の光源を得られる．なお，提案手法では，オフラインで光源分布の取得を行う．まず，鏡面球が観測された画像から，画像中の球の中心と半径を算出する．次に，cube map を用いて球上の全方向のサンプリングを行う．cube map は図 3(a) に示したようなカメラ座標系に対し，立方体によってサンプリング方向を表現する．図 3(b) に立方体の展開図を示しており，各面の法線と一致する軸を示している．本研究では，$32 * 32 * 6$(画素) サイズの cube map を用いる．各画素毎に方向ベクトルを保持しており，これを用いた全方向のサンプリングを行う．各方向ベクトルに対応する鏡面球の画素は次のように算出する．まず，cube map が持つ方向ベクトルと球の中心からカメラへ向かうベクトルの中間ベクトルを算出する．次に，球の法線ベクトルと中間ベクトルが一致する鏡面球の位置を算出する．最後に，算出した位置と対応する画像中の画素の明度値を参照する．cube map の全画素に対して同様の処理を行うことで，cube map で表現された光源分布を得る．

### 2.4 リライティング処理

最後に，陰影を変更する手法及び，その理論について述べる．本研究では，Zhen ら手法 [3] が提案したリライティングのモデルを用いる．無限遠の半径を持つ半球上に光源が存在すると仮定すると，実空間における 3 次元点 $\boldsymbol{x}$ の照度 $E(\boldsymbol{x})$ は次のように表される．

$$E(\boldsymbol{x}) = \int_{\Omega} L(\omega) max(\cos\theta, 0) d\omega \qquad (4)$$

ここで，$L(\omega)$ は，点 $\boldsymbol{x}$ の法線方向に対して $\omega = (\theta, \phi)$ 方向に存在する光源の輝度値，$max(\cos\theta, 0)$ の項は，光源の入射角度による強度，$d\omega$ は点 $\boldsymbol{x}$ からの微小立体角を表している．なお，$\theta$ は仰角，$\phi$ は方位角である．$L(\omega)$，$max(\cos\theta, 0)$ の項はそれぞれ SH に近似 [10] して扱う．ここで，Ramamoorthi ら [11] の表記を用いると，式 (4) は，SH を用いて次のように表される．




(a) カメラ座標系　(b) cube map 座標系

図 3　カメラ座標系と cube map 座標系.
Fig. 3　Camera and cube map coordinates.

$$E(\boldsymbol{x}) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} A_l(\theta) L_{lm} Y_{lm}(\omega) \qquad (5)$$

$Y_{lm}(\omega)$ は SH の基底関数を表している．$A_l(\theta)$ と $L_{lm}$ は，それぞれ $max(\cos\theta, 0)$，$L(\omega)$ の項を SH に近似して得た係数である．本研究では，2.3 節で取得した光源分布を事前に SH に近似して用いる．ここで，Ramamoorthi ら [11] の ronbun では，9 つの SH 基底関数を用いることで，拡散反射による陰影を表現できるという報告がある．そのため，本論文ではバンド数 $l = 2$ とし，9 つの SH 基底関数を用いて近似する．$max(\cos\theta, 0)$ の項は予め $\theta = 0$ の時の値 $A_l^{std} = A_l(0)$ を計算しておき，推定された法線方向に合わせて，$A_l^{std}$ を SH 回転 [12] する事で $A_l(\theta)$ を得る．

対象の物体の反射特性を完全 Lambertian モデルと仮定すると，Src 環境と Dst 環境の光源分布の比を計算する事で，陰影を変更した画像が得られる．

$$i^{dst}(\boldsymbol{u}) = i^{src}(\boldsymbol{u}) \frac{E^{src}(\boldsymbol{x})}{E^{dst}(\boldsymbol{x})} \qquad (6)$$

光源分布の比を取ることで，拡散反射係数をキャンセルでき，反射特性を求めずにリライティングを行える．ここで，$i^{dst}(\boldsymbol{u})$ は画素 $\boldsymbol{u}$ におけるリライティング後の画素値，$i^{src}(\boldsymbol{u})$ は入力画像の画素値，$E^{src}(\boldsymbol{x})$ は Src 環境での光源分布 $L_{lm}^{src}$ によって得られる照度，$E^{dst}(\boldsymbol{x})$ は Dst 環境での光源分布 $L_{lm}^{dst}$ によって得られる照度である．陰影を変更後，Dst 環境の背景に合成する事で，最終結果画像を得る．

## 3 実験

提案手法の有効性を示すために，Src 環境で撮影する対象が動的物体を，動的光源環境へリライティングを行った．図 4 に本実験で用いた，物体と Src 光源環境

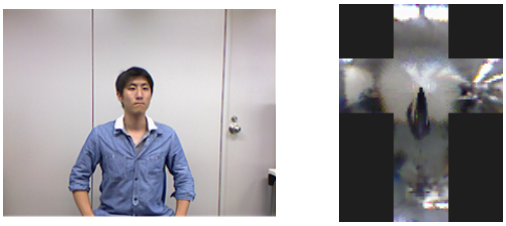


図 4　本実験で用いた物体と Src 光源環境.
Fig. 4　The Object and Src light environment using our experiment.

を示す．本実験んでは，動的物体として人物 **Human** を対象とした．光源環境は図 3(b) の座標系で表される cube map で示している．

次に，実装環境について述べる．本実験では，RGB-D カメラとして Microsoft 社の Kinect を用いた．RGB カメラと距離カメラは同じ位置ではないため，RGB 画像と距離画像に視差が生じる．視差の補正は，RGB-D カメラに関するライブラリー OpenNI [8] を使用した．鏡面球は，半径 5cm のステンレス製の球を用いた．表 1 に実験のパラメータを示す．対象物体までの距離によって画像中の領域の大きさが異なるため，適切なパラメータを設定した．また，本手法では，GPU により並列処理で高速化を行う．ほとんどの処理は，各画素で行う処理を GPU で並列に行うが，k-means クラスタリング，SH 回転，積分画像の処理に関しては，GPU 実装のアルゴリズム [7] [12] [13] を用いた．実装環境は，以下の通りである．OS: Windows7，CPU: Intel Core i7-3940XM 3.00GHz，RAM: 32.0GB，GPU: NVIDIA GeForce GTX 680M，開発環境:Microsoft Visual C++ 2010．カラー画像，距離画像の解像度は 640 × 480 である．

表 1　実験パラメータ.
Table 1　Parameters of experiment.

| 処理名 | パラメータ | **Human** |
|---|---|---|
| マスク作成 | $n_d$ | 10 |
| | $n_c$ | 8 |
| 距離値の補完 [6] | $\sigma_s$ | 1.4 |
| | $\sigma_c$ | 1.2 |
| | $n_s$ | 5 |
| | $t$ | 40 |
| k-means クラスタリング | Cluster number | 20 |
| | Iteration | 20 |
| バイラテラル フィルタ | $\sigma_{s'}$ | 150 |
| | $\sigma_d$ | 60 |
| | $n_{s'}$ | 13 |

### 3.1　実験結果

動的物体として人物 (**Human**) を動的な環境へリライティングを行った結果について述べる．Src 環境は，カメラ上方 ($z+$ 軸方向) に蛍光灯が配置された光源環境となっており，光源によって人物の首の部分に生じている影を除いて，全体の陰影は明るくなっている．Dst 環境は夜間の道路で予め撮影したシーン（**Night Road** と呼称）で，車の往来によって光源環境が動的変化している．光源の変化に応じて，陰影の変化が表現されているかを検証した．なお本処理では，予め撮影した Dst 環境の光源分布と背景画像を毎フレーム更新してリライティング処理を行っている．図 5 にリライティングを行った動画列から 4 つの結果を示す．図 5 では，各フレーム時の光源分布，法線分布，リライティング結果を示している．より結果を見やすくするため，cube map と結果画像の明るさとコントラストを編集した．どのフレーム時に対しても，実験 2 と同様に服のしわや人物の姿勢に合わせて法線分布が推定できていることが分かる．$1^{st}$ フレームと $49^{th}$ フレームでは，カメラ後方から走ってくる車のヘッドライトによって陰影が明るく 表現されている．$159^{th}$ フレームでは，車の通過に伴って光源の位置は対象物体の右横に存在している．そのため，光源と反対方向に法線が向いている領域では暗く，光源の方向に向いている領域は明るい陰影となっている．$222^{th}$ フレームでは，車が完全に通過し，周囲に存在する街灯や信号機の点灯などによって薄暗く陰影が表現されている．この実験より，本手法を用いる事で動的環境に対してリライティングできることが分かる．

最後に処理時間について述べる．表 3 に本実験での各処理時間を示す．距離画像修正と法線分布推定に最も処理がかかっていることが分かる．距離画像修正では k-means クラスタリングを用いており，これは CPU の処理と GPU の処理を交互に行っているためである．また，法線推定手法では，9 つの積分画像を算出しているため，処理が多くかかっている．**Human** の平均処理速度は，10.20fps であった．

表 2　処理時間.
表 3　Computation time.

| 処理名 | **Human** （ミリ秒） |
|---|---|
| 距離画像修正 | 36.51 |
| 領域抽出 | 0.03 |
| 法線分布推定 [9] | 42.69 |
| SH 回転 [12] | 0.01 |
| リライティング [3] | 0.02 |

## 4　おわりに

本稿では，RGB-D カメラを用いた実時間リライティング手法を提案した．本研究は対象の領域抽出，法線分布推定，実時間処理に着目し，それらに対して独自の工夫を施すことにより実時間リライティングを可能

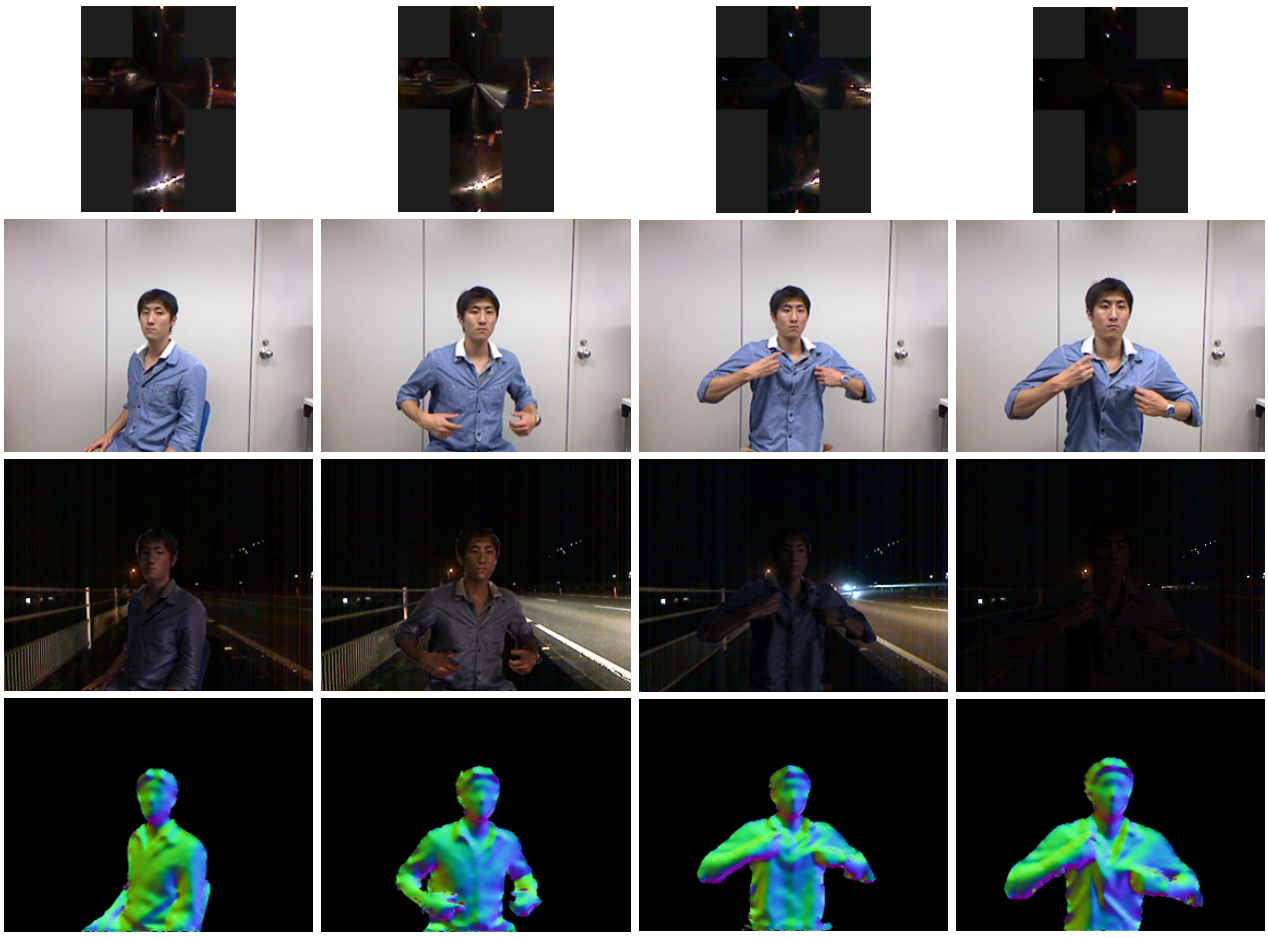



図 5　リライティング結果 (**Human**).
Fig. 5　Relighting results of **Human**.

とした．また，提案手法の処理を GPU による並列演算で高速化する事により，動的環境に対して約 10fps の速度でリライティングを行えることを示した．今後の課題として，鏡面反射を含む陰影や影を考慮したモデルを用いたリライティングの精度向上を目指す．

## 参考文献

[1] http://gl.ict.usc.edu/LightStages/
[2] Paul Debevec "Rendering synthetic objects into real scenes: bridging traditional and image-ased graphics with global illumination and high dynamic range photography", SIGGRAPH, pp. 189-198, 1998.
[3] Zhen Wen, Zicheng Liu and Thomas S Huang "Face Relighting with Radiance Environment Maps", Computer Vision and Pattern Recognition, pp. 158-165, 2003.
[4] Paul Debevec, Tim Hawkins, Chris Tchou and Haarm-Pieter Duiker, Westley Sarokin and Mark Sagar "Acquiring the reflectance field of a human face", SIGGRAPH, pp. 145-156, 2000.
[5] Andreas Wenger, Andrew Gardner, Chris Tchou, Jonas Unger, Tim Hawkins and Paul Debevec "Performance relighting and reflectance transformation with time-multiplexed illumination" ACM Transactions on Graphics 24, 3, pp.756–764, 2005.
[6] Li Chen, Hui Lin, and Shutao Li, "Depth image enhancement for Kinect using region growing and bilateral filter", International Conference on Pattern Recognition, pp. 3070-3073, 2012.
[7] J. Sirotkovic, H. Dujmic, V. Papic, "K-Means Image Segmentation on Massively Parallel GPU Architecture", MIPRO, pp. 489-494, 2012.
[8] OpenNI, http://openni.org/
[9] Stefan Holzer, Radu Bogdan Rusu, M. Dixon, Suat Gedikli, and Nassir Navab, "Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images", IROS, pp. 2684-2689, 2012.
[10] Sloan Peter-Pike, "Stupid Spherical Harmonic (SH) Tricks", Game Developer's Conference, 2008
[11] R. Ramamoorthi, P. Hanrahan, "An efficient representation for irradiance environment maps", SIGGRAPH, pp. 497-500, 2001.
[12] Derek Nowrouzezahrai, Patricio Simari, Eugene Fiume, "Sparse Zonal Harmonic Factorization for Efficient SH Rotation", ACM Transactions on Graphic, 31(3): 23, 2012.
[13] B. Bilgic, B.K.P. Horn, I. Masaki, "Efficient Integral Image Computation on the GPU",In IEEE Intelligent Vehicles Symposium, pp. 528-533, 2010.