

深度カメラによるリライティングを用いた オンライン仮想会議室の実現

田中滉大¹ 高井昌彰² 飯田勝吉² 高井那美³

概要：複数の利用者が異なる光源環境からリモート参加するオンライン会議システムにおいて、実時間のリライティングによって参加者の光源環境を統一化できれば、仮想空間共有における整合性や実在感をより向上させ、新たな高品位インタラクションの実現も期待できる。本研究では Zoom 等のオンライン会議システムをベースとして、参加者の実時間リライティング画像を1つの仮想会議室画像内に描画する仮想会議システムを開発する。参加者の顔及び頭部形状は深度カメラで実時間取得し、ゲームエンジン Unity 上に実装したリライティング処理を別々に施す。仮想光源のオンライン情報は Zoom 画面上の画像データを用いた通信で実現する。

キーワード：オンライン会議システム, 深度カメラ, Unity, リライティング, 仮想会議室

Online virtual conference room with realtime relighting by using a depth camera

KODAI TANAKA^{†1} YOSHIAKI TAKAI^{†2}
KATSUYOSI IIDA^{†2} NAMI TAKAI^{†3}

1. はじめに

リライティングとは仮想光源の照明効果を撮影画像に反映させて、実光源とは異なる照明環境を構築するものである。これまでリライティングには多くの手法が提案されているが、主な手法としては、点光源に照らされた人の顔の反射光を利用したもの[1]、環境マップを利用したもの[2]、深層学習を用いたもの[3]、深度カメラを利用したもの[4]などがある。一般のリライティングでは実物体の3次元形状や表面の反射特性、撮影した環境等を把握していることを前提とすることが多いが、実時間リライティングにおいては、光学的整合性と時間的整合性を高いレベルで両立させることが求められる。

本研究では、オンライン会議システムにリモートから参加する参加者に実時間リライティングを個々に施すことで、全体としての光学的整合性を保ち、仮想空間における参加者の実在感を向上させたオンライン仮想会議室を実現する。参加者の頭部及び顔面画像に対する深度カメラを用いたリライティング処理はゲームエンジン Unity のシェーダーで実装する。またオンライン会議のネットワークそのものは Zoom をベースとして、リライティングされた参加者画像が1つの背景画像内に描画されるオンライン仮想会議室を構築する。その際、参加者画像の画面配置や仮想光源等に

関する参加者間のデータのやり取りは Zoom 画面上の画像情報を介して実現するため、Zoom 以外の特別なデータ通信プロトコルを要しない。本システムをハイエンド GPU 搭載のゲーミング PC 複数台に実装してネットワークを形成することで、オンライン仮想会議室システムとしてのリライティングの効果や描画性能等を実験的に検証する。

2. オンライン仮想会議室システムの概要

オンライン仮想会議室の全体イメージを図1に示す。オンライン会議に参加する各ユーザは深度カメラと RGB カメラが搭載された RGB-D カメラ (Intel RealSense-D415) を利用してオンライン会議に参加する。会議中の参加者画像には共通の背景となる仮想会議室画像の光源環境に近いリライティングが実時間で施され、個々の参加者のローカルな背景を除去した上で、仮想会議室画像上に表示される。

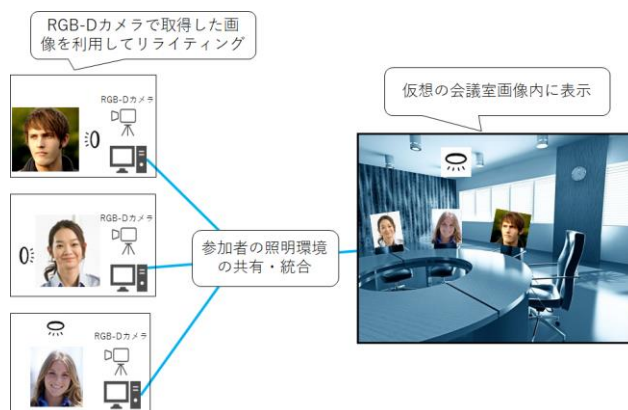


図1 オンライン仮想会議室のイメージ

¹ 北海道大学大学院情報科学院
Graduate School of Information Science and Technology, Hokkaido University
² 北海道大学情報基盤センター
Information Initiative Center, Hokkaido University
³ 北海道情報大学経営情報学部
Hokkaido Information University

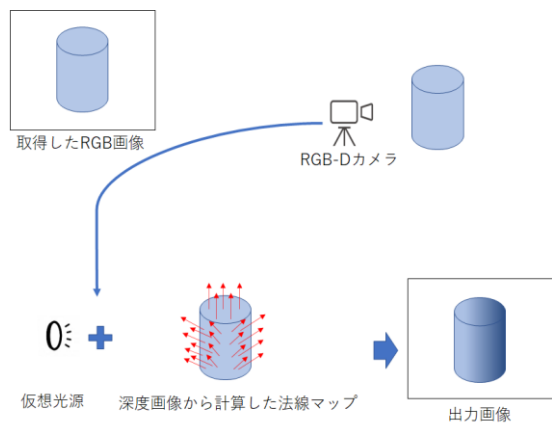


図2 リライティング処理の流れ

オンライン会議参加者の頭部及び顔面の3次元表面形状を実時間で推定するため、本システムでは深度画像を用いる。深度画像を利用したリライティングの流れを図2に示す。RGB-Dカメラを利用することにより、リライティングに必要な色情報と物体表面の深度情報を同時に取得する。

各参加者のPCではRGB-Dカメラから取得した深度画像とRGB画像から物体表面の法線ベクトルを推定し、深度値の計測誤差などによるノイズを除去した後、仮想光源による輝度値を計算することでリライティング画像を出力する。仮想光源の方向は、全参加者共通の仮想会議室背景画像と整合するように定める。

リライティング処理に必要な仮想光源の方向等に関する数値データを、ホスト役となる参加者から他の参加者へと伝達させるため、何らかの通信プロトコルを用いたデータ転送が必要になる。しかしZoomなどのオンライン会議システムのAPIを利用することはバックドア通信に外ならず、ユーザ認証やセキュリティ上の問題が懸念される。そのため本システムでは、Zoom画面上の指定された領域に描画される画像そのものを、符号化された通信データとして利用し、リライティングに必要なデータ転送を実現する。

3. リライティング処理の実装

3.1 シェーダーによるリライティング処理

時間的整合性(対象物体の動きに追従できる処理の高速性)を担保するため、Unityシェーダーを用いてリライティングを実装する[5]。この処理の流れの概要を図3に示す。

最初にRGB-Dカメラから取得した画像にマスク処理を施し、中心差分で求めた勾配ベクトルから法線ベクトルを計算する。次に、25近傍の法線ベクトルを利用したノイズ除去を行い、スムーズシェーディングを実施した結果を利用して法線マップを作成する。法線マップの4近傍の平均化を行った後、仮想光源方向および視点方向から、拡散反射光と鏡面反射光を求め、RGB画像内の対象物体表面の輝度値を変化させる。以上の処理を毎フレームごとに実施することで実時間でのリライティングを実現する。

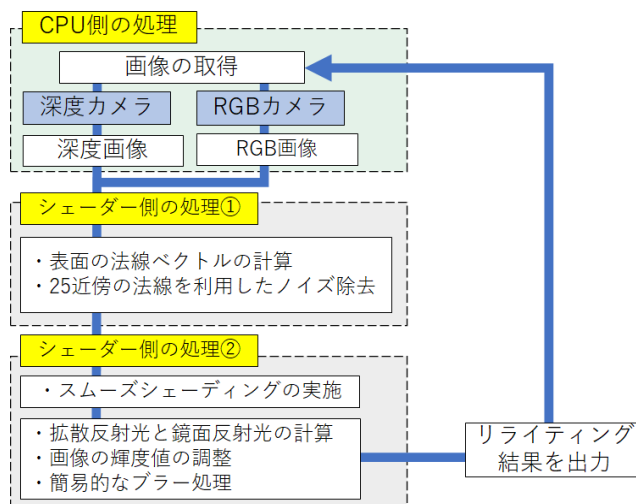


図3 Unityシェーダーによるリライティング

3.2 法線マップ

物体表面の形状を表現する方法は多岐にわたるが、本システムではより計算コストを抑えながら忠実に物体表面を推定できる法線マップを採用した。図4はユーザの顔の深度画像を利用して作成した法線マップである。リライティングの計算時には画像データと同様な行列計算を行うため、シェーダーによる高速な処理を実行することができる。

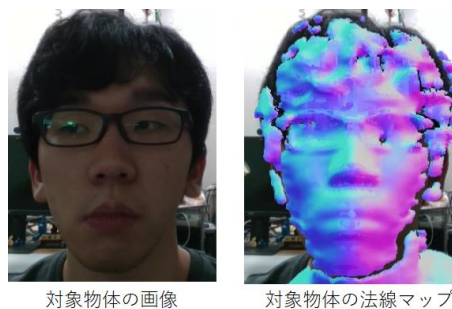


図4 法線マップの例

3.3 マスク処理

オンライン会議参加者の頭部領域だけにリライティングを施すため、あらかじめ深度範囲を定めたマスク処理をおこなう(図5)。領域をしぼり込むことでリライティング処理による遅延を最小化する効果もある。オンライン会議参加者の個々のローカルな背景画像の入れ替えにはZoomが有する機能を利用する。

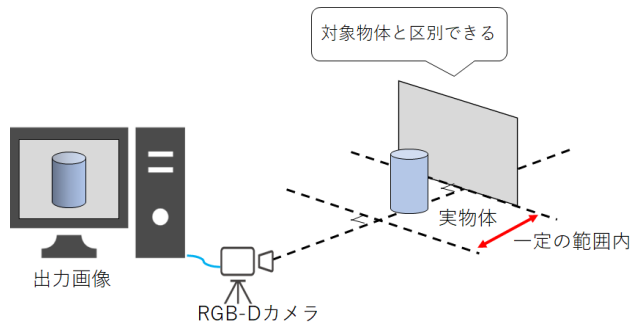


図5 マスク処理

3.4 スムーズシェーディング

対象物体表面の勾配ベクトルから得られる 25 近傍の法線ベクトルを利用した平滑化処理だけでは、リライティングに現れるノイズの除去は必ずしも十分ではない。そこでレンダリングにおけるスムーズシェーディングの考え方を利用する。すなわち物体表面がポリゴンパッチで形成されるものと見なし、隣接する面の法線ベクトルを利用してシェーディングを変化させることで物体表面を滑らかに表現する[6]。取得した深度画像のうち、隣接する同じ深度値を持つピクセルを 1 つの面として仮定し、隣接面の法線ベクトルに対して連続的に値を遷移させる(図 6)。これにより、深度画像に含まれるノイズの影響を低減し、対象物体表面を滑らかに表現することができる。

隣接する同じ深度のピクセルを1つの面としてスムーズシェーディングを実施

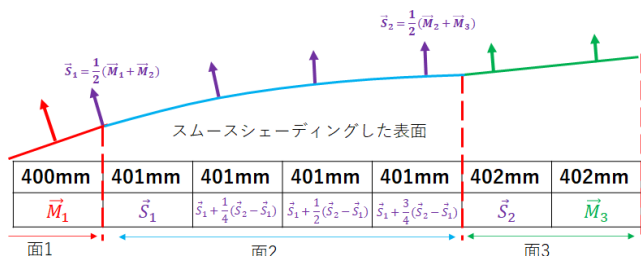


図 6 スムーズシェーディング

3.5 実時間リライティングの結果

ハイエンド GPU 搭載のゲーミング PC (HP OMEN 45L: スペックを表 1 に示す) を用いて実行したカメラ画像のリライティング処理の結果を図 7 に示す。仮想光源の向きの変化に対応したリライティングが実現できていることが分かる。深度カメラ (Intel RealSense-D415) の取得画像サイズ 480×640 ピクセルにおいて、描画フレームレート約 65fps を達成しており、ユーザの顔面画像に対する滑らかなリライティング処理を実時間で実行できている。ただし現状では、対象物体の一部が欠損したり、環境光の影響を十分考慮できていないため、さらなる改善が必要である。

比較のため、異なる CPU/GPU を搭載したミドルクラス PC (Dell Precision Tower5810, CPU: Xeon E5-1607v4, memory 16GM, GPU: NVIDIA Quadro K620) で、リライティング処理を一部簡略化した上で動作させた場合、対象の大きさがほぼ同じ場合でも約 40fps 程度である。リライティングの速度は PC のハードスペックに依存する部分が多い。

表 1 ゲーミング PC のスペック

	HP OMEN 45L	HP OMEN 30L
OS	Windows11 Pro	Windows10 Pro
CPU	Inel Core i9-12900K	Inel Core i9-10850K
GPU	GeForce RTX 3090	GeForce RTX 3090
Memory/SSD	32 GB / 1.86TB	64 GB / 953GB



図 7 リライティング結果

4. オンライン仮想会議室の実装

4.1 リライティング結果の Zoom を利用した統合

オンライン仮想会議室は、共通の背景となる会議室画像の光源環境をもとに施された参加者のリライティング結果を 1 つの会議室画像上で統合することで実現される。

画像をオンラインで実時間共有する手法は多岐にわたるが、本システムでは Zoom と Unity の仮想カメラ機能を利用する。仮想会議室上にリライティング結果を統合する処理の流れを図 8 に、仮想会議室の背景画像を図 9 に示す。

はじめに仮想会議室の背景画像の照明環境に従い、各参加者画像にリライティングを施す。次に個々のリライティング結果を Zoom 会議画面内に参加者画像として表示する。オンライン会議ホスト役のユーザ PC では、Zoom 会議画面内の各参加者画像領域を Unity からキャプチャし、Unity 上の仮想会議室画像内に表示する。その後、参加者のローカル背景画像を仮想会議室の部分領域画像と個々に入れ替え、全体を Unity の仮想カメラを介して Zoom で共有する。

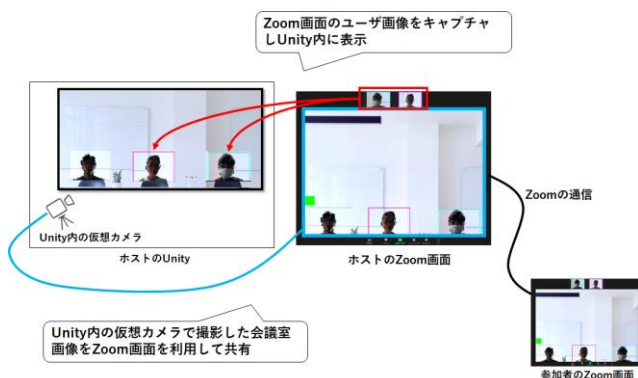


図8 リライティング結果のZoomによる統合

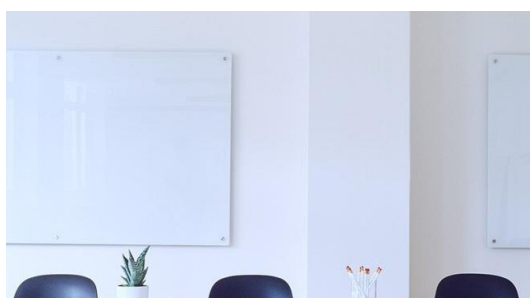


図9 仮想会議室の背景画像

4.2 仮想会議室の配置位置と仮想光源情報の共有

仮想会議室画像内の定められた位置に参加者のリライティング画像を表示するため、会議室画像内の配置場所を簡単な色情報で識別し、場所と参加者を関連付ける。また、仮想会議室画像ごとに照明環境(仮想光源の強さ・方向等)を予め設定し、これを参加者間で共有する。

仮想会議室画像内に参加者が表示される位置に紐づく識別色情報は会議開始前にホストから全参加者に配布される。会議開始後、各参加者は配布された識別色画像を表示し、ホスト側ではZoom会議画面上でそれぞれの識別色が表示されている画像領域を抽出する。その後、この領域座標をもとに参加者画像をUnityからキャプチャし、対応した領域に参加者のリライティング画像を表示する。

仮想光源方向についても同様な手法で共有が可能であるが、一般的には座標位置を2次元コードで符号化し、これを表示・認識すること情報共有を行うことが考えられる。

4.3 Zoomを利用した通信の結果

参加者3名のオンライン会議を想定し、本システムの動作実験を行った。参加者のZoom画面全体を図10に示す。また、比較のためリライティング前後の参加者画像領域の部分拡大を図11に示す。仮想会議室の背景画像に見られる映り込みや陰影の様子から、部屋の左側面に位置する窓から部屋右奥に向かう入射光が想定されるが、その光源方向と光学的に整合するように参加者個々のリライティング処理が施され、参加者の実在感が向上していることがわかる。

オンライン会議ホスト役はゲーミングPC(HP OMEN 45L)を利用したが、約55fpsの処理速度を達成している。他参加者のPCは前述のミドルクラスPC(Dell Precision T5810)とゲーミングPC(HP OMEN 30L)である。ミドルクラスPCでは処理速度約26fps、ゲーミングPCでは約45fpsを達成している。ミドルクラスPCでは実行速度が十分ではないため、処理のさらなる軽装化が必要である。



図10 オンライン仮想会議室の動作結果



(a)リライティング前



(b)リライティング後(仮想光源:水平に左から)

図11 仮想会議参加者に対するリライティングの効果

5. まとめと今後の課題

Unityを開発プラットフォームに、深度カメラで取得された人物頭部・顔面の表面形状をもとに、実時間のリライティング処理を実現するシェーダーを構築した。また、このリライティングをオンライン会議システムZoomに適用し、複数の参加者画像を配置した会議室全体の光学的整合性を有するオンライン仮想会議室を実現した。ハイエンドクラスのゲーミングPCを用いて、人物像に対するリライティングを実施したところ、約65fpsの処理速度を達成できることが分かった。また、Zoomベースのオンライン仮想会議室の実装では約55fpsを実現できることが分かった。

オンライン仮想会議室の完成度・実用性を高めるには、参加者画像の解像度の向上やマスク処理の精度の向上に加え、個々の参加者のローカルな環境光を考慮したリライティングの質向上が必要であり、これらを処理の実時間性の制約のもとに実装することが今後の課題である。

参考文献

- [1] P. Debevec, T. Hawkins, C. Tchou H. P. Duiker, W. Sarokin and M. Sagar, “Acquiring the reflectance field of a human face”, *SIGGRAPH*, pp. 145-156, (2000)
- [2] Z. Wen, Z. Liu and TS. Huang, “Face Relighting with Radiance Environment Maps”, *IEEE Computer Society CVPR*, pp. 158-165, (2003)
- [3] Deep single image portrait relighting,
<https://zhoper.github.io/dpr.html>
- [4] 池田拓也, 小倉洋平, ドゥ ソルビエ フランソワ, 齋藤英雄:
“RGB-D カメラを用いた実時間リライティング映像生成システムの開発”, 映像情報メディア学会誌, Vol.68, No.12, pp. J558-J568 (2014)
- [5] 田中滉大, 高井昌彰, 飯田勝吉, 高井那美: “深度カメラを用いた低遅延リライティングの Unity 実装”, 令和3年度電気・情報関係学会北海道支部連合大会講演論文集, no.120 (2021)
- [6] 田中滉大, 高井昌彰, 飯田勝吉, 高井那美: “深度カメラを用いたリライティングの画質向上”, 令和3年度電子情報通信学会総合大会講演論文集, D-11-24, (2022)