Vol.2017-CVIM-207 No.7
2017/5/10

深層学習による集光模様の実時間生成

岡本 拓朗[1,a)]　浦西 友樹[1,2,b)]　間下 以大[2,c)]　ポチャラ ラサミー[2,d)]　清川 清[3,e)]　竹村 治雄[1,2,f)]

概要：プロジェクションディスプレイ技術を用いた物体の見かけを制御する質感操作の研究は，テーマパークや美術館といった場所でメディアアート作品へ活用されている．物体の質感のうち，インテリアとしても利用価値の高いサンキャッチャーやガラス細工といった光を反射・屈折する透明物体の質感を付与すれば鑑賞者を楽しませることが出来るのではないかと考えた．本研究では透明物体を通過した反射光・屈折光が背景に衝突することで現れる集光模様に着目する．集光模様の特性のひとつに物体が移動することで影のようにその様子が変化することが挙げられる．しかし，光の遮蔽によって生じる影とは異なり，移動する物体の集光模様を再現する場合，光源との位置関係の変化による光線の屈折・反射の軌道の再計算を行わなければならず，既存の大域照明が計算可能なアルゴリズムでは処理時間が問題となる．そこで，対象物体の位置，形状の情報から背景に投影されるであろう集光模様画像を深層学習によって生成する手法を提案する．本手法では畳み込みニューラルネットワーク (Convolutional Neural Network, CNN) に深度画像と集光模様画像の対応関係を学習させた．実験では処理時間の計測と出力画像の品質について，既存のレンダリングソフトウェアとの比較，検証を行った．

キーワード：質感操作，透明物体，畳み込みニューラルネットワーク，深層学習，集光模様

1. はじめに

色彩や陰影，光沢だけでなく，物体が持つ透明感も質感の一部である．光を屈折・反射する透明物体の質感はサンキャッチャーや水晶玉，ガラス細工のようにインテリアやアート作品に多く用いられている．この透明物体の質感を不透明物体に付与出来れば鑑賞者に強い印象を与える効果が期待できる．天野 [1] はプロジェクタとカメラを同軸光学系として構築し，質感操作対象の投影画像の入手のような前処理や，投影画像の位置合わせの作業を不要にする手法を提案している．この手法では，カメラより取得した操作対象の画像に処理を加えて投影することで，物体表面の色彩やコントラストの操作，透明感や光沢感を付与を可能にしている．しかし，実際の透明物体には環境中に存在する光源による光が反射，屈折することで生成される集光模様が床や壁に生じるはずである．透明物体である人工溶錬水晶玉および不透明物体である発泡スチロールに光を当てたときの様子を**図 1** に示す．物体の大きさ，形状はほぼ等しいが，図 1(a) のように透明な水晶玉が置かれている床には集光模様が現れている．天野の手法 [1] では背景に投影される集光模様のように物体表面以外の操作については言及していない．

(a) 透明物体の影　　(b) 不透明物体の影

図 1　透明物体と不透明物体の影の比較

集光模様は光の物理法則に従って映し出されるため，物体の形状や位置，光源の種類，屈折率などのパラメータによって一意に決定される．既存のレンダリング手法のうち，計算量の面で優れている光線追跡法 [2] では直接光の計算を行わないため集光模様を描画できない．有限要素ラジオシティ法 [3] では間接光の表現に優れているが，対象となる物体が拡散面であることを想定したアルゴリズムのため，集光模様の描画には不適である．集光模様の描画が可能な経路追跡法 [4] やフォトンマッピング法 [5] ではノイズ除去や膨大なフォトンの計算が必要となる．レンダリン

[1] 大阪大学 情報科学研究科
[2] 大阪大学 サイバーメディアセンター
[3] 奈良先端科学技術大学院大学 情報科学研究科
[a)] okamoto.takurou@lab.ime.cmc.osaka-u.ac.jp
[b)] uranishi@ime.cmc.osaka-u.ac.jp
[c)] mashita@ime.cmc.osaka-u.ac.jp
[d)] photchara@ime.cmc.osaka-u.ac.jp
[e)] kiyo@ime.cmc.osaka-u.ac.jp
[f)] takemura@ime.cmc.osaka-u.ac.jp

グ時に多大な処理時間が要求される場合，物体と光源の位置関係が変化したときに形状が変わる集光模様の特性を再現する際に問題となる．

ある入力が与えられたときに特定のパターンに従ってデータを出力する手法のひとつにニューラルネットワークを用いる手法が存在する．特に画像群を入力に与えるネットワークの場合，Lo ら [6] が提案した，2 次元の重みパラメータによって学習する畳み込み層を組み込んだ畳み込みニューラルネットワーク（CNN）がしばしば用いられる．Simo-serra らの研究 [7] ではラフスケッチを自動的に変換する手法で CNN を用いている．畳み込み演算によって入力に与えられたラフスケッチの解像度を下げ，小さな特徴マップに圧縮する処理を行い，逆畳み込み演算によって元の解像度に復元し，画像変換を実現する．この手法では入力画像とサイズの同じ線画の画像を出力するため解像度の低下による画質の劣化が生じない．また，画像入力から出力までの計算時間にも優れており，解像度が 1024 x 1024 の画像でも 1 秒に満たない処理時間を実現している．

本研究では，実時間で実行できる集光模様のレンダリング方法として深層学習を用いた手法を提案する．提案手法では，ユーザの操作が加わる体験型インタラクティブアートのような，ユーザの操作に即座に反応できる即応性が求められるようなシステムで使用されることを想定する．

2. 深層学習を用いた集光模様生成

2.1 システム概要

本手法は想定される特定の実空間に光源と移動可能な実物体を設置し，その実物体が透明物体であると仮定したときに，光源によって背景に投影されるであろう集光模様を実時間で再現する．提案システムの概要図を**図 2** に示す．スクリーンに影が投影されるよう光源とスクリーンの間に透明感を付与したい不透明物体を設置する．その物体の位置や形状を深度カメラから深度画像として取得し，学習済みのニューラルネットワークに入力する．深度画像の特徴を識別し，スクリーンに投影されるであろう集光模様画像を出力する．その集光模様画像をスクリーンの該当位置に重畳投影することで，影の操作を実現する．

2.2 機械学習による集光模様の生成

環境内に光源と透明物体，および集光模様が投影される背景が存在するとき，その背景に投影される集光模様は以下の条件で決定される．

(1) 物体の形状
(2) 物体，背景および光源の位置関係
(3) 仮定した材質によって決定される屈折率，透過率，色
(4) 仮想光源の強さ，方向

ここで，仮想光源の性質，透明物体の材質ををあらかじめ定義したとき，未知のパラメータは物体の形状および背景

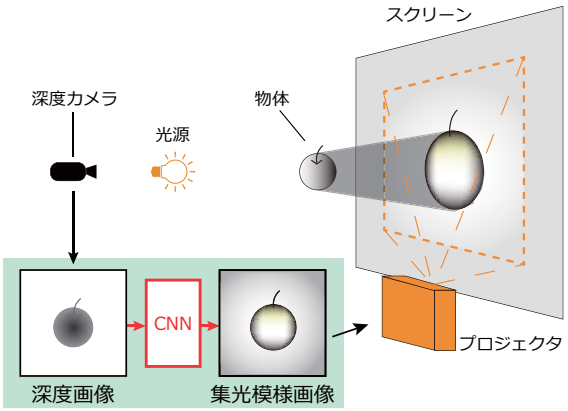

図 2 提案システム概要図

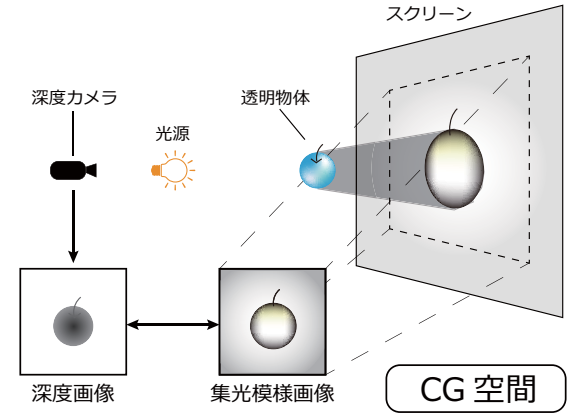

図 3 データセット生成時の CG 空間の様子

との位置関係だけになる．背景の位置，仮想光源の位置と性質を定義し CG 空間にその環境を再現する．CG 空間内に実世界同様深度カメラを設置し，その空間内に存在する物体の位置と形状を深度画像として出力する．また，その物体がある特定の材質と特性をもった透明物体と仮定したとき，背景に生成される集光模様の画像を記録する．この CG 空間の様子を**図 3** に示す．この一連の処理を様々な形状の物体を CG 空間内で位置や回転を変えながら行い，深度画像と集光模様画像の組を大量に用意する．これらの画像群 (以下，データセット) を，深度画像を入力データ，集光模様画像を目標出力としてニューラルネットワークに学習させることで，深度画像から読み取れる物体の形状，位置の特徴と集光模様の特徴をニューラルネットワークによって対応付けさせる．このようにして，物体の形状，位置を深度カメラで計測し，得られた深度画像から集光模様を生成するモデルを作成する．

本手法では深度画像を 0〜255 の画素値をもつグレースケールの画像とする．光源は白色光，透明物体の材質を無色のガラスに設定し，分光は生じないものとする．生成される集光模様は深度画像と同様に 1 チャンネルのグレースケールで描画を行う．

表 1　計算機の仕様

項目	詳細
OS	Windows 10 Enterprise 64-bit
CPU	Intel Core i7-6700K CPU @ 4.00GHz (8 CPUs)
メモリ	32.0GB
GPU	NVIDIA GeForce GTX 1080

2.3 畳み込みニューラルネットワーク

深度画像から読み取れる物体の表面の形状，スクリーンまでの距離，位置といった各画素値の相関関係を活用するため，本ニューラルネットワークでは CNN を用いる．出力画像の品質を考慮すると，入力で与える深度画像よりも解像度を下げることは好ましくない．また，入力よりも高い解像度で出力するのは画質の低下を引き起こす恐れがある．そこで，Simo-serra らの手法 [7] で用いられた CNN を参考にし，最初は画像を小さな特徴マップに圧縮して処理を行い，その後元の解像度に復元し入力画像と同じ解像度の画像を出力するネットワークを採用する．入力に与える画像の解像度を $(H \times W)$ としたときの構築したネットワークの構造を**図 4** に示す．入力画像と同じ解像度を出力 (flat-convolution)，あるいは解像度を縮小させる役割を果たす層 (down-convolution) では畳み込み層を，解像度を拡大させる役割を果たす層 (up-convolution) では逆畳み込み層を使用している．また，出力層以外の各層を通過するごとに出力されるデータに対して ReLU(Rectified Linear Unit) 関数を適用させる．出力層にはシグモイド関数を適用する．学習の指標に用いる損失関数には平均二乗誤差（Mean Square Error, MSE）を採用する．モデルの出力と目標出力の画素に対して MSE を計算し，その値が小さくなるよう勾配を求めパラメータを更新する．勾配の更新方法には AdaGrad[8] を用いている．

3. 試作システム

3.1 実装環境

提案手法を実装した試作システムを python3.5.2 を用いて実装した．また，ニューラルネットワーク実装用ライブラリの chainer[9] を用いて CNN を構築し，学習後のモデルを生成した．**表 1** にシステムを構築した計算機の仕様を示す．

3.2 データセット生成方法

深度画像と集光模様のデータセットをレンダリングソフトウェアの POV-Ray(Ver3.6)[10] を用いて作成した．POV-Ray のシーン中にカメラ，光源，集光模様を投影する背景としてスクリーンを設置する．各要素の位置関係を**図 5** に示す．空間内には $x = 0$ の位置にスクリーンとして白色の無限平面，座標 (1,0,0) にカメラおよび光源を設置している．カメラには一般的なパースペクティブカメラを，光源には白色の点光源を採用した．カメラのレンズは原点方向に向いており，水平画角は 70°である．物体は $(0 \leq x \leq 1)$ を満たし，かつ物体がカメラやスクリーンに接触せず，カメラで捉えられる範囲をランダムに移動する．集光模様の描画アルゴリズムにはフォトンマッピング法を利用した．

3.2.1 学習データ

学習対象に 8 種類の形状の物体（立方体，直方体，球体，円筒，円錐，円盤，トーラス，板状の直方形）を選んだ．これらの形状の物体を 1 つ，位置および回転をランダムに変化させながら POV-Ray のシーン中に描画し，1 形状あたり 5000 組，計 40000 組の深度画像および集光模様画像を出力した．位置・回転の変動には疑似乱数を使用し，各シーンに再現性を持たせている．**図 6** に各形状の深度画像（上段）及び集光模様画像（下段）の例をに示す．

3.2.2 未知形状データ

学習結果のモデルの評価に用いるために，学習データには組み込まない形状の物体を用意した．各種の形状は，円筒と球を組み合わせたダンベル型，円筒の片方の面を除去し内側を空洞にしたコップ型，学習データで用意した直方体と同じ形状で内側を空洞にしたケース型，球を二つ接着させた雪だるま型，正八角柱型の 5 種類である．各物体の位置及び回転をランダムに変化させて 1 形状あたり 20 組，計 100 組のデータセットを用意した．

4. 実験と考察

4.1 学習結果

3.2.1 節で述べた学習用データセットを用いてニューラルネットワークの学習を行った．データセットの解像度は全て (128×128) で統一した．3.2.1 節で述べたようにランダムに物体の位置および回転を変更したため，深度画像に物体が一切表示されない，すなわち深度画像の全画素が 255 である画像がいくつか確認された．それらのデータは学習データに使用せず除外した．除外されていない学習データのうち，約 1 割の 3980 組のデータセットをテスト用データセットとして抽出し，残りの 35826 組を訓練用データセットとして利用した．

訓練用データセットを計 100 回 (100 エポック) 学習させた．1 エポック経過ごとにテスト用データをモデルに読み込ませ，損失関数の値を記録し学習度合を確認した．n エポックの学習終了時の重みパラメータが記憶された学習モデルを model n と定義する．バッチ数を 10 に設定したところ，テスト用データセットによる評価も含めた 1 エポックごとの処理時間は約 13 分，100 エポック完了するのにかかった処理時間は約 1300 分だった．訓練用データ，学習用データの損失関数の値を**図 7** に示す．図 7 より，学習を重ねるにつれて損失関数の値が小さくなっていることから，勾配が正常に更新されパラメータが最適値に向かっていることが読み取れる．従って，学習に成功していると判

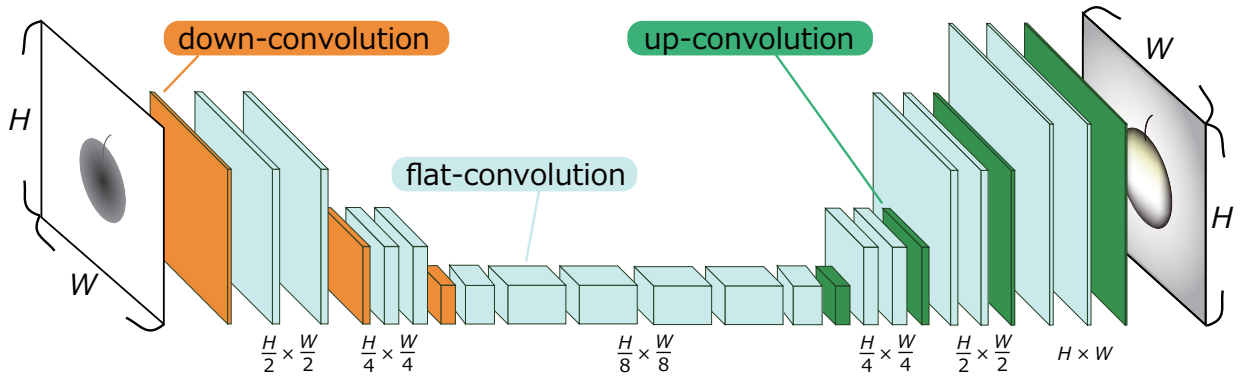

図 4　構築したネットワーク

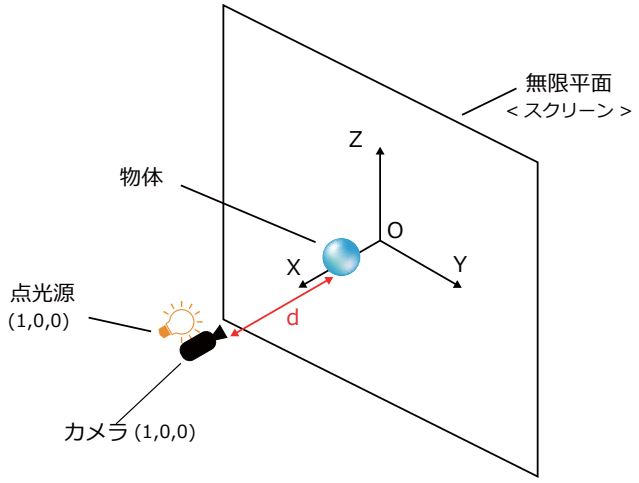

図 5　POV-Ray 内のカメラ，光源，スクリーンの位置関係

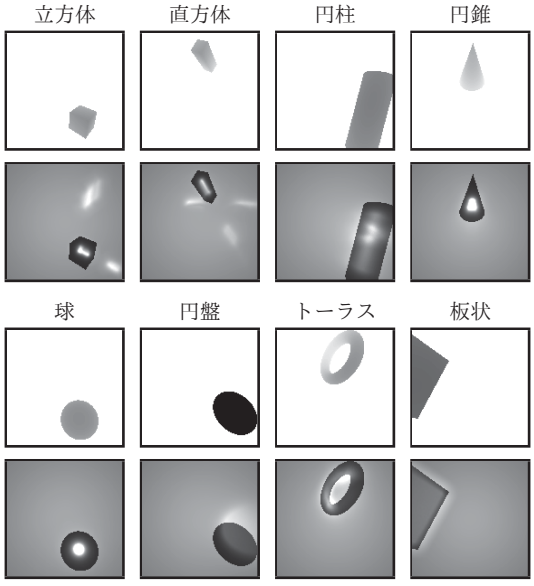

図 6　学習用データセットの例

断した．

訓練用およびテスト用データセットから深度画像と集光模様画像の組をランダムにそれぞれ 100 組抽出した．これらのデータセットと 3.2.2 節で述べた未知形状データセット 100 組を次節以降の画像類似度，処理時間の計測時に使用する．

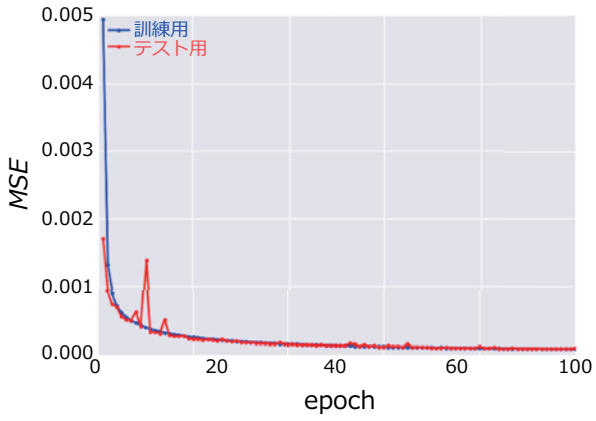

図 7　学習曲線

4.2　画像類似度の計測

集光模様の再現度を調べるために，本手法で作成した集光模様画像と目標出力に指定した POV-Ray の集光模様画像の類似度を計算した．

4.2.1　実験設定

4.1 節で抽出した訓練用，テスト用データセット各 100 組および未知形状データセット 100 組の深度画像を model 1，model 5，model 20，model 100 にそれぞれ入力して集光模様画像を出力した．

本手法で生成した集光模様の品質を評価するために，本手法で生成した集光模様画像と，POV-Ray で出力した集光模様画像との画像類似度を，各画素の差の二乗和の総和で比較する SSD (Sum of Squared Difference) で計算した．データセットで目標出力に指定した集光模様画像を Ground Truth(以下，GT) とみなし，これらとモデルから出力された集光模様画像とを比較する．出力画像を Y，GT を Y^* としたとき，$f_{SSD}(Y, Y^*)$ は式 1 で表される．

$$f_{SSD}(Y, Y^*) = ||Y - Y^*||^2{}_{FRO} \quad (1)$$

訓練用，テスト用，未知形状の各データセット 100 組に対して $f_{SSD}(Y, Y^*)$ の値を求め，その平均と標準偏差を記録

表 2　$f_{SSD}(Y, Y^*)$ 平均

モデル	訓練用	テスト用	未知形状
model1	118.53	103.99	423.24
model5	43.000	33.347	464.88
model20	17.700	14.339	457.98
model100	6.6973	6.5374	464.24

表 3　$f_{SSD}(Y, Y^*)$ 標準偏差

モデル	訓練用	テスト用	未知形状
model1	99.393	74.295	388.85
model5	60.663	21.748	419.34
model20	26.555	11.283	408.21
model100	4.1980	4.5640	421.06

した．

4.2.2　結果

出力結果の一例として，テスト用及び未知形状データセットを model 100 に入力した際の出力結果と比較対象の GT をそれぞれ**図 8**，**図 9** に示す．各データセットを各モデルに入力した際の出力結果に対する $f_{SSD}(Y, Y^*)$ の値の平均を**表 2**，標準偏差を**表 3** に，グラフを**図 10** に示す．

4.2.3　考察

図 10 より，学習回数が 100 エポックに近づくにつれて訓練用，テスト用データセットにおいて $f_{SSD}(Y, Y^*)$ の値が小さくなることが読み取れる．図 8 の出力結果より，学習データに直接組み込んだか否かに関わらず，その形状のデータセットを学習させていれば高い類似度を得られることが確認された．しかし，未知形状のデータセットに対しては学習を重ねても類似度の向上は見られなかった．特に図 9 の 2，3 列目の出力結果から読み取れるように，ガラスコップや中空の直方体などの中空の物体は深度画像からその情報を読み取ることが出来ず，それぞれ学習データに組み込んだ，形の近い円柱や密な直方体だと判断し，集光模様画像を生成したと考えられる．中空の物体以外にも，学習に組み込んだ形状を組み合わせて作成した雪だるま型，ダンベル型も輪郭が歪む，集光地点が複数生成されるなどといった曖昧な集光模様が出力されることが確認された．

図 10 より，学習データに使用した形状の物体に対してのみ高い類似度を計測したことから，特定のデータにしか対応しない過学習が発生している可能性が疑われる．この特性は，本手法の利用時にあらゆる物体を対象にする場合には不都合な特性となるが，対象物体の形状が既知の場合には優位に働くと考えられる．

4.3　処理時間の計測

4.3.1　実験設定

訓練用，テスト用，未知形状のデータセットからランダムに 100 組抽出してモデルに入力し，集光模様画像を出力した際の各処理時間を記録した．モデルには最も学習が進んでいると考えられる model 100 を使用した．ここで，処理時間は入力に使用する画像を指定してから画像を画像ファイルとして出力するまでの時間と定義する．比較対象として，POV-Ray で各形状が均等に含まれるよう，学習データからランダムに 100 組選択し，レンダリングした際の 1 画像当たりの平均処理時間を記録した．

表 4　処理時間 平均

項目	訓練用	テスト用	未知形状	POV-Ray
処理時間 (sec)	0.01616	0.01606	0.01654	0.4956

表 5　処理時間 標準偏差

項目	訓練用	テスト用	未知形状	POV-Ray
標準偏差	0.004902	0.003910	0.003760	-

4.3.2　結果

各条件での平均処理時間を**表 4**，標準偏差を**表 5** に示す．ただし，POV-Ray には各画像ごとの処理時間を出力する機能は備わっていないため，標準偏差は記録していない．

4.3.3　考察

表 4，表 5 より，フォトンマッピングを利用している POV-ray では実時間処理を達成できていないが，本提案手法の処理時間は実時間処理の目安となる約 0.033sec (30 fps) を下回り，実時間処理が可能であることが示された．提案手法では入力画像の種類に問わず計算量が一定のため，データセットの種類ごとの平均処理時間，標準偏差にほとんど差はなく，安定した処理時間で集光模様画像が得られることが確認された．この特徴によって，実環境に対して本手法を適用したときに複雑な形状の物体を対象にしても処理時間が変わらず，安定したフレームレートでシステムを動作させることが可能であると考えられる．

5.　おわりに

本研究では深度画像を入力することで，実時間でスクリーンに投影される集光模様画像を出力するニューラルネットワークを開発した．また，評価実験より，学習データに含んだ形状の物体の深度画像を入力に与えた場合には目標出力に非常に近い画像を出力することが確認された．本手法の使用が考えられる，実時間処理を要するインタラクティブアートや映像演出の場面で，対象となる物体の形状が既知である場合，事前に対象となる物体のデータセットで学習を行うことで高品質な集光模様を実時間で得られることが期待される．学習データに組み込んでいない物体の深度画像に対しては学習を繰り返しても再現性は向上しなかったものの，反射光や集光らしき様子が一部の画像で確認された．

今後の課題としては，本手法の汎用性を高めるために，あらゆる形状の物体を対象とする必要がある場合にも対応できるようネットワーク，データセットを工夫することが挙げられる．また，実際に実世界の深度画像を入力し，影

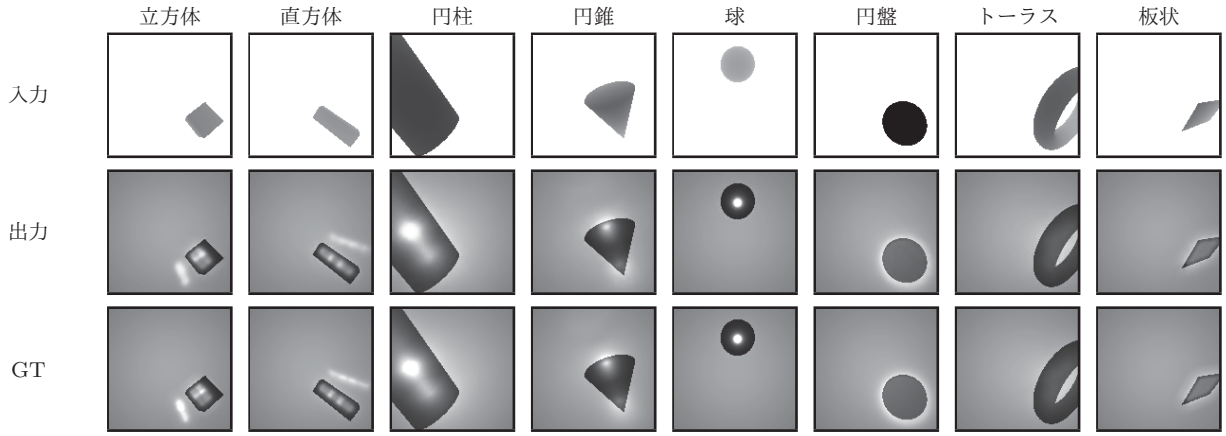

図 8　テスト用データセット 出力結果の例

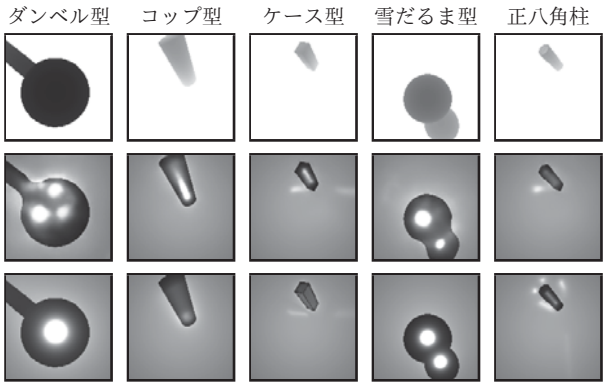

図 9　未知形状データセット 出力結果の例

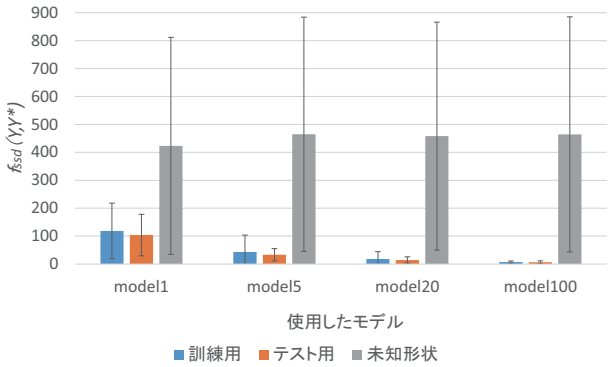

図 10　$f_{SSD}(Y, Y^*)$ の平均と標準偏差

の質感操作を行うことで鑑賞者に強い印象を与えることが出来るか検証するために，影の実時間操作を行うシステムの実装を進めていきたい．

謝辞　本研究は JSPS 科研費 16H02858 の助成を受けたものである．

参考文献

[1] 天野 敏之，“プロジェクタカメラ系を用いた光沢感と透明感の実時間操作”, 映像情報メディア学会誌, Vol. 68, No. 12, pp. 528 - 533, 2014.

[2] Turner Whitted, “An improved illumination model for shaded display,” *Communications of the ACM,* Vol. 23, No. 6, pp. 343 - 349, Jun 1980.

[3] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg and Bennett Battaile, “Modeling the interaction of light between diffuse surfaces,” In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques (SIGGRAPH ‘84)* Vol. 18, No. 3, pp. 213 - 222, Jul 1984

[4] Jmes T. Kajiya, “The rendering equation,” In *Proceedings of ACM SIGGRAPH Computer Graphics,* Vol. 20, No. 4, pp. 143 - 150, Aug 1986.

[5] Henrik Wann Jensen 著, 苗村 健 訳 “フォトンマッピング 実写に迫るコンピュータグラフィックス”, オーム社, Jul 2002.

[6] Shih-Chung B. Lo, Heang-Ping Chan, Jyh-Shyan Lin, Huai Li, Matthew T. Freedman and Seong K. Mun, “Artificial convolution neural network for medical image pattern recognition,” *Neural Networks*, Vol. 8, No. 7, pp. 1201 - 1214, 1995.

[7] Edgar Simo-serra, Satoshi Iizuka, Kazuma Sasaki and Hiroshi Ishikawa, “Learning to Simplify: Fully Convolutional Networks for Rough Sketch Cleanup,” ACM Transactions on Graphics (TOG) In *Proceedings of ACM SIGGRAPH 2016,* Vol. 35, No. 4, p. 121, Jul 2016.

[8] Duchi, John, Elad Hazan and Yoram Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” In *Proceedings of Journal of Machine Learning Research 12,* pp. 2121 - 2159, Jul 2011.

[9] Chainer: A flexible framework for neural networks `http://chainer.org/`, 最終アクセス 2017 年 4 月 12 日

[10] POV-Ray - The Persistence of Vision Raytracer `http://www.povray.org/`, 最終アクセス 2017 年 4 月 12 日