Vol.2022-CG-186 No.4
2022/6/27

微分可能レンダリングによる ガウシアンフィルタリングのパラメータ推定

北林悠河 [1]　岩崎慶 [1]

概要：近年，並列計算機の急速な発展を背景に，機械学習をレンダリングパイプラインに取り込む形式で特殊化した，微分可能レンダラと呼ばれるレンダリングパラメータ推定手法が注目されている．Laine らは，プリミティブと呼ばれる処理単位を定義し，それらを組み合わせてパイプラインを構築する，モジュール式の微分可能レンダリングパイプラインを提案した．これにより，柔軟性・汎用性に優れた微分可能レンダラの構築を可能とした．本稿では，Laine らが提案したモジュール式の微分可能レンダリングパイプラインを拡張し，ガウシアンフィルタのフィルタ強度を推定する手法を提案する．連続的なパラメータでガウシアンフィルタのフィルタ強度を定義し，勾配法によって高精度に学習が行えることを示す．また，他のレンダリングパラメータ学習とガウシアンフィルタとの親和性についても検証する．

キーワード：微分可能レンダリング，ガウシアンフィルタ

1. はじめに

2D 画像からレンダリングパラメータを推定するインバースレンダリングの一つである微分可能レンダラは，機械学習における勾配法をレンダリングパイプラインに取り入れた推定手法である．つまり，微分可能レンダラは，パイプラインによって生成される画像を目標画像に近づけるようにレンダリングパラメータ学習を行うアルゴリズムである．微分可能レンダラという構想自体は新しいものではないが，ここ数年の並列計算機の性能向上と機械学習の発展に伴って注目されてきた背景がある.微分可能レンダラは，コンピュータグラフィックスにおけるパラメータ表現の明瞭性と，機械学習のパラメータ学習の効率性を合わせ持ったパラメータ推定が期待できる．

Laine らは，プリミティブと呼ばれる処理単位を組み合わせて，微分可能レンダリングパイプラインを構築する手法[1]を提案した．微分可能レンダリングの従来手法では，特化したパイプラインによる特定の学習対象のパラメータ推定を行なっており，柔軟性・汎用性には重点を置いていない．あらゆるレンダリング手法に対してパラメータ学習を可能にするためには，柔軟性・汎用性のある微分可能レンダラが必要とされる．

Laine らが提案したモジュール式微分可能レンダリングパイプラインは，ユーザ独自のパイプライン構築を可能とした，拡張性に優れた設計となっている．また，パイプラインに用いられるパラメータは，微分可能なパラメータであれば，誤差逆伝播法に基づいた勾配定義が可能であるため，レンダリングパラメータの自由な学習対象設定が可能となる．しかしながら，Laine らの手法では，被写界深度のような後処理に対応していないという問題がある．

本稿では，Laine らが提案したモジュール式微分可能レンダリングパイプラインを拡張し，Gaussian Filtering プリミティブの提案と評価を行う．被写界深度への応用の第一歩として，Gaussian Filtering の標準偏差（フィルタ強度）を微分可能レンダリングにより推定するためのプリミティブを提案する．提案法を Laine らのモジュール式微分可能レンダリングパイプラインに組み込むことによって，オブジェクトの頂点座標，頂点色，およびフィルタ強度を同時に学習できた．

2. 関連研究

2.1 微分可能レンダラ

Kato らは，ラスタライズが離散的な処理であるという主張のもと，ラスタライズのための近似勾配を利用し，レンダリングをニューラルネットワークに統合させた Neural 3D Mesh Renderer (NMR)[2]を提案した．Liu らも同じ主張のもと，ラスタライズにおけるポリゴンの存在を確率的なものとして捉えた連続関数を定義し，メッシュ形状推定を行う SoftRas[3]を提案した．しかし，これらのアプローチは，正しい勾配を定義していないうえ，パイプラインに柔軟性を求めていない．

Li らは，モンテカルロレイトレーシング法による物理ベース微分可能レンダリング[4]手法を提案した．Nimier-David らは，メッシュ形状のみならず，多用途にパラメータ推定を行うことができるフレームワーク，Mitsuba2[5]を提案した．これらの手法は，性能面において大量学習には向いていないという問題がある

Laine らが提案したモジュール式微分可能レンダリングパイプライン[1]は，パイプラインを再利用可能な処理単位に分け，モジュール式に組み立てるといった微分可能レンダラ構築手法である．このシンプルなアプローチにより，正しい勾配計算のもと柔軟性・汎用性に富んだ高性能なパイプライン構築を可能とした．

1 和歌山大学
Wakayama University

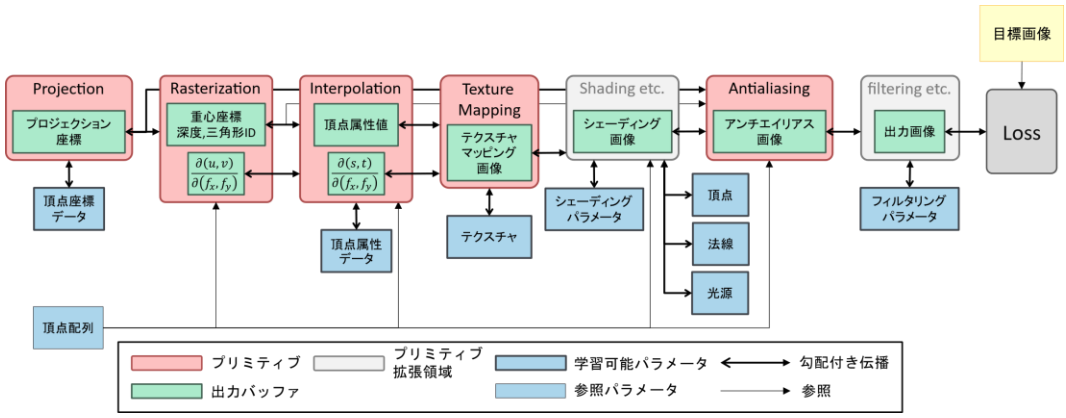

図 1　パイプライン全容図

2.2　プリミティブ

モジュール式微分可能レンダリングパイプラインの構築には，そのパイプラインに使用されるプリミティブの定義が要求される．それぞれのプリミティブでは，フォワードパスとバックワードパスを定義する必要がある．フォワードパスでは，そのプリミティブにおける入力パラメータから出力パラメータを得る演算を定義する．これは，通常のレンダリングパスに相当する処理である．バックワードパスでは，フォワードパスの演算における入力パラメータごとの勾配値を定義する．

パイプラインは，プリミティブを組み合わせて構築される．パラメータ依存関係にあるプリミティブにおいて，フォワードパスおよびバックワードパスの接続が行われる．フォワードパスではレンダリングパスを形成する処理手順で接続され，バックワードパスでは，フォワードパスの逆順を追う形で，損失関数の勾配値を逆伝播させる．

2.3　学習パラメータ

フォワードパスを通じて参照されるレンダリングパラメータには，多様なパラメータが存在する．微分可能なパラメータであれば，バックワードパスを通じて損失関数の勾配値を取得することが可能であるため，勾配法による学習が可能となる．頂点，オブジェクト姿勢，頂点色をはじめとした，基本的なレンダリングパラメータは当然ながら学習可能である．また，テクスチャマップ，光源，反射特性などのパラメータは，演算モデルによって処理内容が異なるが，モデルに応じてフォワードパスとバックワードパスを正しく定義すれば学習可能である．

パラメータの学習には，目標画像と予測画像との差を表す損失関数と，損失を最小化させるためのパラメータ最適化アルゴリズムを使用する．Laine らは，損失関数に二乗和誤差(Sum of Squared Error，(SSE))，を使用し，最適化アルゴリズムに Adam[6]を使用した．

3.　パイプライン全容

図 1 は，モジュール式微分可能レンダリングパイプラインの全容図である．プリミティブ間のパラメータ依存関係が図示されており，学習パラメータから出力画像をレンダリングするフォワードパスと，出力画像から学習パラメータへの逆伝播を促すバックワードパスが示されている．フォワードパスにおけるプリミティブの出力は，後続のプリミティブの入力パラメータとして接続される．もしくは，最終出力であれば，予測画像として目標画像と比較され，損失関数を通して損失を得るために利用される．損失関数の勾配値は，最終出力を担ったプリミティブのバックワードパスに接続される．バックワードパスにおいては，フォワードパスでパラメータの接続関係にあったプリミティブ間で，後続のプリミティブより，そのパラメータに関する損失関数の勾配値が伝播される．この作用により，誤差逆伝播を行うことができる．つまり，自プリミティブの入力パラメータに関する損失関数の勾配値を定義するためには，微分の推移律に従って，逆伝播されてきた損失関数の勾配値と自プリミティブで定義した勾配値を乗算すればよい．

3.1　基本プリミティブ

基本プリミティブは，開発背景より，Projection プリミティブ，Rasterization プリミティブ，Interpolation プリミティブの 3 つとした．これらのプリミティブは，3D レンダリングを行う上で，必須となるプリミティブである．

Projection プリミティブは，座標変換を担うプリミティブである．主には，頂点座標の投影変換を取り扱う．必要によって，頂点座標のビューポート変換や法線ベクトルの回転など，投影変換ではない座標変換も取り扱う．フォワードパスで頂点座標からプロジェクション座標への変換を行う場合，バックワードパスでは，逆伝播されてきたプロジェクション座標に関する損失関数の勾配値に，プロジェク

ション座標を頂点座標で偏微分した値を乗算することで，頂点座標に関する損失関数の勾配値を求めることができる．

Rasterization プリミティブは，ラスタライズ処理を担うプリミティブである．頂点配列より 3 頂点を結んでできるスクリーン座標上の三角形内において，3 頂点のプロジェクション座標から描画ピクセルごとの重心座標を求める．フォワードパスでプロジェクション座標から重心座標を求めているため，バックワードパスでは，逆伝播されてきた重心座標に関する損失関数の勾配値に，重心座標をプロジェクション座標で偏微分した値を乗算することで，プロジェクション座標に関する損失関数の勾配値を求めることができる．

Interpolation プリミティブは，重心座標に基づいて，頂点属性値の線形補間を行うプリミティブである．頂点属性とは，頂点が保有する様々な数値情報で，頂点色や頂点法線などが該当する．また，テクスチャを利用する場合，テクスチャ UV 座標も頂点属性として処理される．重心座標は描画ピクセルごとに計算されているため，頂点属性値も描画ピクセルごとに割り当てられる．フォワードパスで重心座標から頂点属性値を求めているため，バックワードパスでは，逆伝播されてきた頂点属性値に関する損失関数の勾配値に，頂点属性値を重心座標で偏微分した値を乗算することで，重心座標に関する損失関数の勾配値を求めることができる．

3.2 Antialiasing プリミティブ

Laine らは，頂点の学習を促すべく，モデルの幾何情報を参照し，アンチエイリアシングを行う Antialiasing プリミティブを提案した．具体的には，三角形 ID の異なる隣接ピクセル間において，三角形の辺が横切る位置によって，移す色量を決定するアンチエイリアシング手法である．この手法は，DEAA[7]や GPAA[8]といったアンチエイリアシング手法を派生させた手法で，ラスタライズの連続化を行う処理であるとも言える．

Antialiasing プリミティブのバックワードパスでは，逆伝播されてきた出力色に関する損失関数の勾配値より，入力色に関する損失関数の勾配値に加え，頂点のプロジェクション座標に関する損失関数の勾配値を求めることができる．

4. 提案法

ガウシアンフィルタは，画像にぼかしを施す処理として代表的なフィルタリング処理であり，被写界深度表現などにも応用されるフィルタリング処理である．我々は，フィルタ強度をパラメータ化したガウシアンフィルタリングを提案する．フォワードパスにおける畳み込み演算の式を以下に表す．

$$\hat{c}_{i,j} = \sum_{x,y} G(\sigma, x, y)\, c_{i+x,j+y} \tag{1}$$

ここで，ピクセル位置(i,j)において，$\hat{c}_{i,j}$は出力色であり，$c_{i,j}$は入力色である．(x,y)は，ピクセル位置(i,j)における相対ピクセル位置である．式(2)に用いられるフィルタ係数Gは，以下の式で表される．

$$G(\sigma, x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2}$$

式(2)のσは，本来，正規分布における標準偏差を示す値であるが，ここではフィルタ強度と見なすことができる．

ここで，バックワードパスでは，逆伝播されてきた出力色に関する損失関数の勾配値より，入力色$c_{i,j}$に関する損失関数$Loss$の勾配値と，学習パラメータとしてのフィルタ強度σに関する損失関数の勾配値が計算され，以下の式のように表される．

$$\frac{\partial Loss}{\partial c_{i,j}} = \sum_{x,y} \frac{\partial Loss}{\partial \hat{c}_{i-x,j-y}} \frac{\partial \hat{c}_{i-x,j-y}}{\partial c_{i,j}} = \sum_{x,y} \frac{\partial Loss}{\partial \hat{c}_{i-x,j-y}} G(\sigma, x, y) \tag{3}$$

$$\begin{aligned} \frac{\partial Loss}{\partial \sigma} &= \sum_{i,j}\sum_{x,y} \frac{\partial Loss}{\partial \hat{c}_{i-x,j-y}} \frac{\partial \hat{c}_{i-x,j-y}}{\partial G(\sigma,x,y)} \frac{\partial G(\sigma,x,y)}{\partial \sigma} \\ &= \sum_{i,j}\sum_{x,y} \frac{\partial Loss}{\partial \hat{c}_{i-x,j-y}} c_{i,j} \frac{\partial G(\sigma,x,y)}{\partial \sigma} \end{aligned} \tag{4}$$

式(4)においてフィルタ係数Gをσで微分した値は，以下の式で表される．

$$\frac{\partial G(\sigma,x,y)}{\partial \sigma} = \frac{x^2+y^2-2\sigma^2}{2\pi\sigma^5} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{5}$$

式(4)にて計算されたフィルタ強度に関する損失関数の勾配値をもって，フィルタ強度への最適化が行われる．

5. 実験結果

図 2 は，今回の実験に使用するパイプライン構成を示している．実験では，頂点位置を$[-1/2, 1/2]^3$，頂点色を$(R,G,B) = [0,1]^3$とする色付き立方体を使用した．このオブジェクトを目標オブジェクトとする．学習オブジェクトの頂点位置は，目標オブジェクトの頂点位置よりさらに$[-1/2, 1/2]^3$の範囲でランダムにずらした位置を初期値として与え，頂点色は$[0,1]^3$の範囲でランダムな値を初期値として与えた．目標画像の生成には，予測画像の生成と同じ構成のパイプラインを使用する．学習において，オブジェクトはランダムな姿勢をとる．実行環境として，Intel Core

i7-10700 CPU，16GB RAM，NVIDIA GeForce RTX 3060 Ti GPU を搭載した PC を使用した．

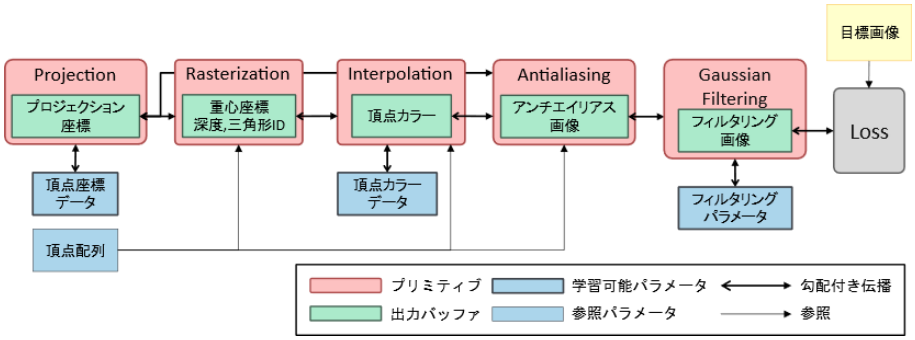

図 2　実験に使用したパイプライン図

5.1　頂点座標・頂点色の学習

まず，ガウシアンフィルタリングを使用しないパイプラインで，色付き立方体の頂点座標と頂点色の学習実験を行う．これは，Laine らが行っていた実験の追実験であり，Antialiasing プリミティブの有用性を測る実験として行う．アンチエイリアシング(AA)処理対象ピクセルが適度に分散される 8×8，16×16，32×32 の解像度で，Adam のハイパーパラメータは頂点座標・頂点色ともに $\beta_1 = 0.9, \beta_2 = 0.999$，学習率$\alpha = 10^{-3}$として学習を進めた．

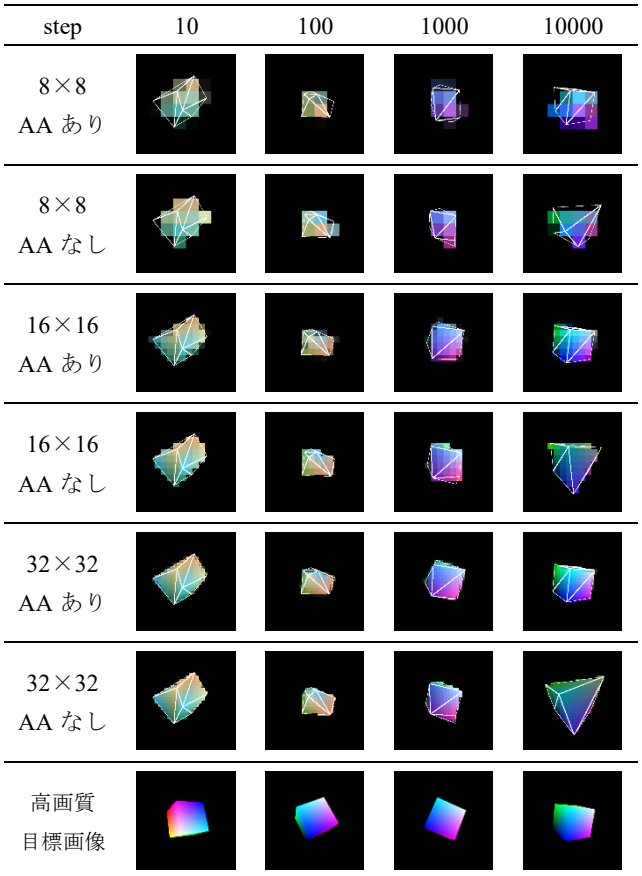

図 3　頂点座標・頂点色の学習の様子

図 3 は，8×8，16×16，32×32 のそれぞれの解像度で Antialiasing プリミティブの有無による頂点座標・頂点色の学習の様子を示した図である．この時，32×32 の解像度において，フォワードパス・バックワードパス・最適化にかかった平均時間は，アンチエイリアシングなしのパイプラインでは 1.4ms，アンチエイリアシングありのパイプラインでは 2.0ms であった．

Antialiasing プリミティブを採用していない学習においては，頂点座標の学習が正しく行えていない．Antialiasing プリミティブでは，特に背景とオブジェクトとの境界において正しい境界線へと誘導しているため，スクリーン全体としては正しいシルエットを得る働きをしていると考えられる．

5.2　フィルタ強度の学習

次に，Gaussian Filtering プリミティブのフィルタ強度の学習実験を行う．実験では，色付き立方体を推定対象のオブジェクトとした．フィルタ強度としての学習パラメータの初期値 $\sigma = 2$ とし，目標値を初期値より小さく設定した $\sigma = 1, 1.4$の場合と，目標値を初期値より大きく設定した $\sigma = 2.8, 4$の場合の学習を行った．32×32 の解像度で，オブジェクトの姿勢は固定させ，Adam のハイパーパラメータは $\beta_1 = 0.9, \beta_2 = 0.999$，学習率$\alpha = 10^{-2}$として学習を進めた．

図 4 は，それぞれの目標値におけるフィルタ強度の学習の様子を図示したものである．この時，フォワードパス・バックワードパス・最適化にかかった平均時間は，目標値 σ=1 とした場合で 1.7ms，目標値 σ=4 とした場合で 2.4ms であった．フィルタ強度の学習は大小の方向によらず，推定することができることが示された．

5.3　頂点座標・頂点色・フィルタ強度の学習

最後に，頂点座標・頂点色の学習に加え，同時にフィル

タ強度を学習する実験を行う．32×32 の解像度で，Adam のハイパーパラメータは$\beta_1 = 0.9, \beta_2 = 0.999$，頂点位置および頂点色では，学習率$\alpha = 10^{-3}$とし，フィルタ強度では，学習率$\alpha = 10^{-2}$として学習を進めた．

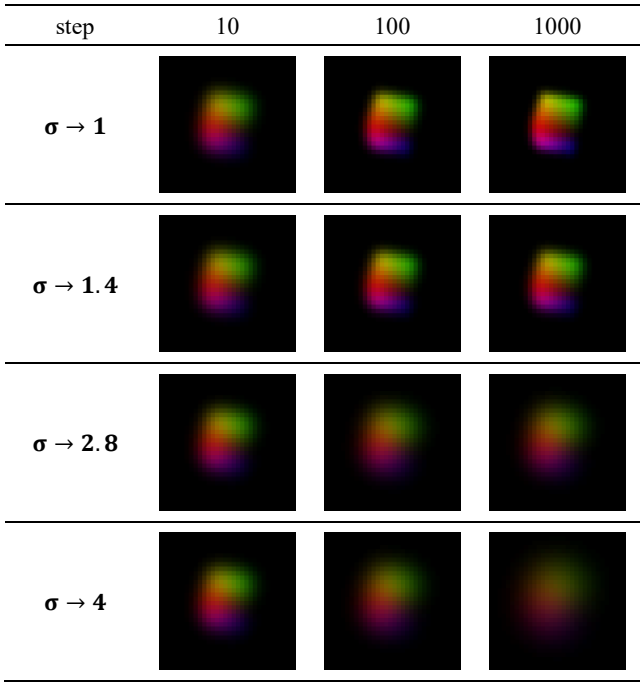

図 4　フィルタ強度の学習の様子

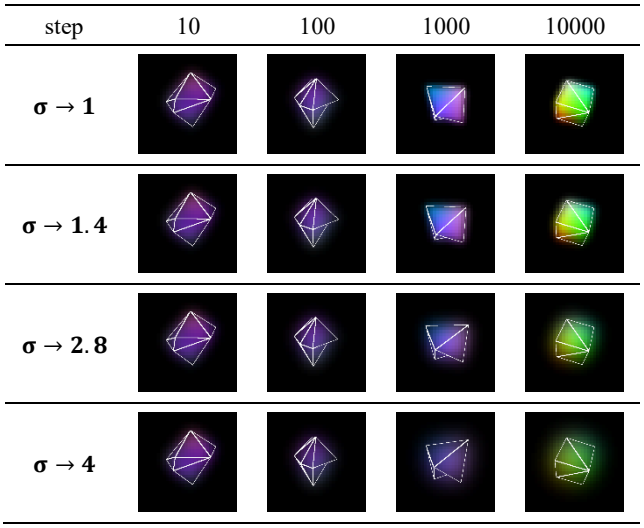

図 5　頂点座標・頂点色・フィルタ強度の学習の様子

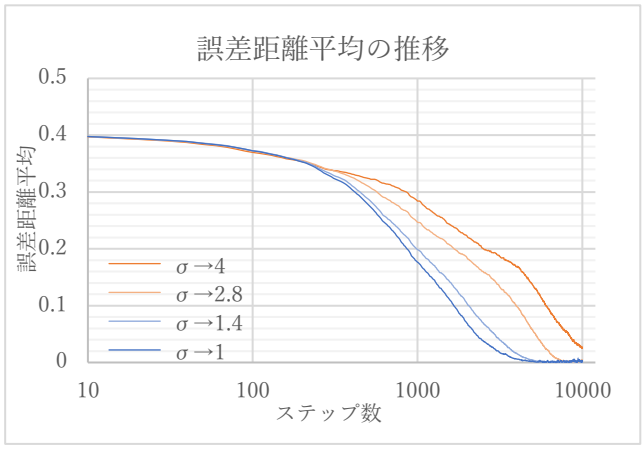

図 6　オブジェクト頂点の誤差距離平均推移グラフ

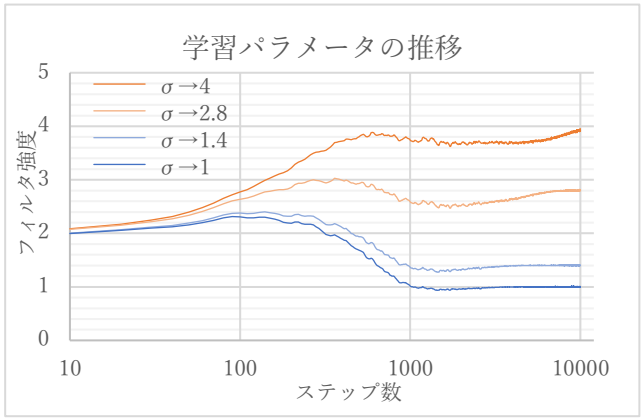

図 7　フィルタ強度としての学習パラメータの推移グラフ

図 5 は，それぞれの目標値における頂点座標・頂点色・フィルタ強度の学習の様子を図示したものである．頂点座標をわかりやすく表示するために，ワイヤーフレームでも表示している．図 6・図 7 は，それぞれ，立方体の 8 頂点の誤差距離平均とフィルタ強度としての学習パラメータの学習ステップごとの推移のグラフである．図 6 に示されるように，4 種類のフィルタ強度全てにおいて，目標とする頂点との誤差距離平均が 0 に近づいていることから，Gaussian Filtering によるぼかしを考慮しつつ，頂点位置を学習していると言える．また，図 7 に示されるように，10,000 ステップにおいて目標となるフィルタ強度にそれぞれ収束していることがわかる．この時，フォワードパス・バックワードパス・最適化にかかった平均時間は，目標値 σ=1 とした場合で 1.7ms, 目標値σ =とした場合で 2.9ms であった．フィルタ強度・頂点位置・頂点色の学習を同時に進めているにも関わらず，いずれも高精度に推定が行えている．

6.　まとめ

本研究では，モジュール式の微分可能レンダリングパイ

プラインの構想に基づいて，Gaussian Filtering プリミティブを提案し，目標画像からフィルタリング強度を学習する実験を行った．それにより，単純な形状ではあるものの，勾配法に基づいて，頂点座標・頂点色・フィルタ強度といったレンダリングパラメータを同時に推定した． Gaussian Filtering プリミティブでは，画像のぼかし具合をパラメータ化することで，そのぼかし具合を学習対象にできることが分かった．

今後の課題として，提案法を発展させた被写界深度表現の実装が挙げられる．リアルタイムレンダラでは，被写界深度表現のようなレンズ効果によるぼかし処理に，高速で円形なガウシアンフィルタを用いられることが多い．微分可能レンダラにおいても，レンダリング画像の写実性と大量学習を行うための高速性が要求される．そのため，提案法を発展させてそのどちらの要件も満たした被写界深度表現を行うことが今後の課題である．また，被写界深度表現では，深度に基づいてフィルタ強度が計算されるため，フィルタ強度から深度への逆伝播が可能であると考えられる．

参考文献

[1] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, Timo Aila, “Modular primitives for high-performance differentiable rendering”, ACM Transactions on Graphics, Vol. 39, No. 6, Article No. 194, pp. 1-14, 2020.
[2] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada, ” Neural 3D Mesh Renderer”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3907-3916, 2018.
[3] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li, ”Soft Rasterizer: A Differentiable Renderer for Image-based 3D Reasoning”, In ICCV, pp. 7708-7717, 2019.
[4] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen, “Differentiable Monte Carlo Ray Tracing through Edge Sampling”, ACM Transactions on Graphics, Vol. 37, No. 6, Article No. 222, pp. 1-11, 2018.
[5] Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob, ”Mitsuba 2: A Retargetable Forward and Inverse Renderer”, ACM Transactions on Graphics, Vol. 38, No. 6, Article No. 203, pp. 1–17, 2019.
[6] Diederik P. Kingma, Jimmy Ba, “Adam: A Method for Stochastic Optimization”, In ICLR, 2015.
[7] Jorge Jimenez, Diego Gutierrez, Jason Yang, Alexander Reshetov, Pete Demoreuille, Tobias Berghoff, Cedric Perthuis, Henry Yu, Morgan McGuire, Timothy Lottes, HughMalan, Emil Persson, Dmitry Andreev, Tiago Sousa, “Filtering approaches for real-time Antialiasing”, SIGGRAPH '11: ACM SIGGRAPH 2011 Courses, Article No. 6, pp. 1-329, 2011.
[8] Emil Persson, “Geometric Post-Process Antialiasing”, http://www.humus.name/index.php?page=3D&ID=86, 2011.