

スペクトル類似度を考慮した深層学習による RGB 画像からスペクトル画像への変換手法

坂本 真啓^{1,a)} Raytchev Bisser¹ 玉木 徹¹ 金田 和文¹

概要: スペクトル画像は工業、医療、リモートセンシング等の幅広い分野にわたり活用されている。コンピュータグラフィックスでもスペクトラルレンダリングにおけるライトプローブ画像やテクスチャとして利用されている。しかし、スペクトル画像の取得には装置や時間的なコストがかかり、その取得や利用の妨げとなっている。本研究では深層学習を用いることにより RGB 画像からスペクトル画像への変換を行う手法を提案する。従来の深層学習を用いたスペクトル画像変換では、RGB 画像とスペクトル画像を直接エンドツーエンドで学習を行う方法を採用していた。これに対し提案手法では輝度値を直接学習するのではなく、スペクトル類似度を損失関数としてネットワークを学習させる。これによりスペクトル画像変換の精度向上をはかる。2つのデータセット (CAVE, VISTA) を用いて RGB 画像からスペクトル画像変換を行い、提案手法の有用性を示す。

Spectral reconstruction from RGB Image using CNN with spectral similarity

MASAHIRO SAKAMOTO^{1,a)} RAYTCHEV BISSER¹ TORU TAMAKI¹ KAZUFUMI KANEDA¹

Abstract: Spectral images are used in a wide range of fields such as industry, medicine, and remote sensing. They are also used in computer graphics as a light probe image and textures in spectral rendering. However, the acquisition of spectral images is costly in terms of equipment and time, which hinders its acquisition and use. In this study, we propose a method for converting RGB images into spectral images by using deep learning. Conventional spectral image reconversion using deep learning has adopted a direct end-to-end learning method for RGB images and spectral images. In contrast, our proposed method does not learn the luminance values directly, but rather learns the network using the spectral similarity as a loss function. This will improve the accuracy of spectral image conversion. We show the usefulness of the proposed method by performing spectral image reconstruction from RGB images using two datasets (CAVE, VISTA).

1. はじめに

RGB 画像は人間の視覚特性から赤、青、緑を用いて色を表現しているのに対して、スペクトル画像は数十以上ある多数の波長帯 (バンド) を含む画像である。狭い波長間隔、つまりバンド数が多いほど、対象物のスペクトル情報を細かく豊富に表現できる。自然界にある物質は固有のスペクトルを持っており、スペクトル画像は対象物の特性を

より詳細に反映する。豊富なスペクトル情報により、人間の視覚やバンド数の少ない RGB 画像では困難な識別が可能であり、工業、医療、農業、リモートセンシング等の幅広い分野にわたり活用されている [1], [2]。また、物理ベースのレンダリングシステムの多くは物理的に正しく光のシミュレーションや画像生成を行うため、スペクトル画像を用いたレンダリングが必要となる [3]。

一般的なカメラのセンサが比較的広い波長帯をいくつか取得するのにに対して、スペクトル画像を取得するハイパースペクトルカメラは狭い範囲で細かく波長を識別し数十から数百バンドの波長情報を取得できる。しかし、狭い波長

¹ 広島大学 大学院先進理工系科学研究科
Graduate School of Advanced Science and Engineering,
Hiroshima University

^{a)} masahiro-sakamoto1@hiroshima-u.ac.jp

帯で画像を撮影すると、画像に関する多くの情報を取得できるが、1つのセンサーに到達する光量も比例して減少するので、解像度を低く抑えなければ良質な信号雑音比を維持するのが難しくなる [4]。したがって、スペクトル画像の取得には空間解像度とスペクトル解像度のトレードオフが問題となる。また、特殊なセンサーが必要とされ、複雑な処理が要求されることからカメラのコストが高いといった問題点もある。

このため、スペクトル画像の超解像に関する研究が盛んに行われてきた。スペクトル画像の超解像は空間解像度とスペクトル解像度の2種類に分けることができる。本研究ではスペクトル解像度に対する超解像に焦点をあて、バンド数が少ないRGB画像から同じ空間解像度で多数バンドを持つスペクトル画像への生成を取り扱う。多くの空間超解像の研究分野 [5] と同様に、畳み込みニューラルネットワーク (CNN) を用いた深層学習を行うことで、RGB画像からスペクトル画像へのマッピングを推定する。

深層学習を用いた既存のアップサンプリング手法では、各スペクトル強度値を直接出力することで画像を生成する。それに対して本研究ではスペクトル強度値を直接求めるのではなく、スペクトル類似度を考慮した変換手法を提案する。出力されたスペクトル類似度に対して適切なスケールリングを行うことで、元のRGB画像の輝度と再構成したスペクトル画像の輝度は一致する。生成結果についてはスペクトル類似度の向上ではなく、真値画像との誤差が小さくなることを目標にする。また、スペクトルの表現方法を複数用意し、その結果についても検討する。

2. 関連研究

2.1 レンダリングを目的としたアップサンプリング手法

カラーディスプレイへの表示にはRGBの三波長成分のみで十分かもしれないが、光と物質の正確なシミュレーションには不十分である。また、物理ベースのレンダリング技術はリアリスティックな表現を可能とするが、可視光域にわたって密にサンプリングされた波長成分が必要となる。しかし、多くの一般的なテクスチャデータがRGB表色系で表現されている。したがって、これらからスペクトラルレンダリングを行うために、RGBの三刺激値から等価なスペクトル分布に近似させる研究が以前から行われてきた。

スペクトラルレンダリングは物理的に正しい画像を生成できるが、サンプリングされる波長成分が多くなるのでレンダリング時の計算時間やメモリ使用量が大きくなってしまいう問題点がある。そのため、スペクトルは効率的に計算できる表現で構成した方が望ましい。また、通常は波長の変化に対してスペクトル強度の変化は滑らかである。そのため、基底関数を用いて、かつ少ない係数でスペクトルを再構成する手法が提案されている [6], [7]。

2.2 スペクトル画像の超解像

スペクトル画像の超解像として既存の研究の多くは空間解像度の復元を試みており、それに対してスペクトル情報の復元に関する研究は少ない。

スペクトル画像の空間解像度を向上させるためには、高解像度のRGB画像と低解像度のスペクトル画像を組み合わせる手法がある。この手法は、画像間の位置を正確に合わせて取得しなければならないが、2種類の画像を適切に撮影することは困難を伴う。この解決策として、いくつかの研究ではRGBカメラのスペクトル応答を用いることで、空間上の誤差を是正した [8]。

これに対して、事例ベースの超解像は1枚の低解像度画像から高解像度画像を復元する。この分野は、近年発達してきたCNNを用いた深層学習が精度を向上させていることから盛んに研究が行われている。その研究成果に触発され、ネットワークモデルが学習するRGB画像をスペクトル画像に置き換えて考えることも可能であり、高い空間解像度を持つスペクトル画像の生成にCNNを適用した手法が提案されている [9]。

RGB画像からスペクトル画像に変換する事例ベースの超解像は本研究と最も関連がある。空間情報についての超解像とは異なり、スペクトルに対する超解像は画像の空間解像度を下げることなくスペクトル解像度における高周波成分を復元する。これに関連する問題としてグレースケール画像をカラー画像に自動変換する手法が挙げられる。これは1つのバンドからRGBの3バンドへとアップサンプリングすることに他ならない。最近の手法ではCNNを使用し、画像の輝度と色度を分けて推測することで従来より優れた結果を得る手法も提案されている [10]。RGB画像からスペクトル画像への変換には、RBF (放射基底関数) ネットワークを用いた手法 [11] や、K-SVD を使用してスペクトル画像の辞書を学習する手法 [12] などが提案されている。また、Galliani ら [13] は Semantic Segmentation に用いられたCNNをスペクトル超解像に適用した。

3. Fully Convolutional DenseNets

本章では、提案手法で用いる深層学習のネットワークモデルについて概説する。Galliani ら [13] の研究と同様に、本研究ではJégou ら [14] によって提案されたモデルの改良を行った。Jégou らは Semantic Segmentation の問題に対処するために Densely Connected Convolutional Networks (DenseNet) [15] を Fully Convolutional DenseNets へと拡張した。

3.1 DenseNets

従来のCNNでは、 $l-1$ 番目の層の出力は l 番目の層の入力となる。 x_l を l 番目の層の出力とすると、 x_l は次のように表される。

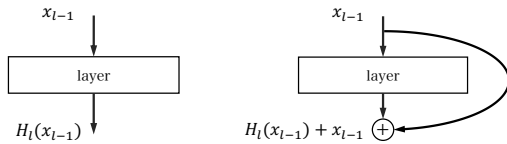


図 1 通常の CNN(左図) と Res block(右図) の比較
Fig. 1 Comparison of standard CNN (left) and Res block (right)

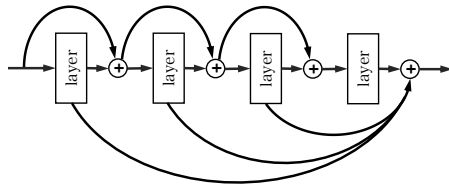


図 2 Dense block の構成
Fig. 2 Dense block configuration

$$x_l = H_l(x_{l-1}) \quad (1)$$

ここで、 H は非線形な関数である。主に畳み込み演算が行われ、通常は活性化関数がそれに続いて適用される。

しかし、単純に層を深くすると勾配が消失してしまい、性能が悪化する問題がある。He ら [16] は入力値を参照した残差関数を学習する方が、それを参照しない従来の非線形関数を学習するよりも容易だと仮定し、ResNet[16] を考案した。ResNet では層の入力値を shortcut connections によって、層の出力値と足し合わせることで残差関数を学習する (図 1 参照)。これにより、勾配の減衰を防ぐことができ、非常に深い層を持つネットワークの学習が可能になる。結果の出力 x_l は次のようになる。

$$x_l = H_l(x_{l-1}) + x_{l-1} \quad (2)$$

層間の情報伝達をさらに改善するため、DenseNet はある層から後続の全層を密に連結した (図 2 参照)。つまり、 l 番目の層は入力として、それより前の層での出力 x_0, x_1, \dots, x_{l-1} を受け取る。 l 番目の層の出力は以下のようになる。ここで $[\dots]$ は要素の連結を表す。

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}]) \quad (3)$$

各関数 H_l が k 個の特徴マップを生成する場合、 l 番目の層には $k_0 + k \times (l - 1)$ 個の特徴マップが入力される。ここで、 k_0 は Dense block への入力となる特徴マップ数である。通常、 k は比較的小きな値 ($k = 12, 16$) などに設定される。また、特徴マップ数が少しずつ増加することから、 k はネットワークの growth rate と呼ばれる。growth rate は各層が特徴マップに追加する新しい情報の量をどの程度にするかをコントロールする役割がある。

3.2 構成要素

提案された FC-DenseNets の構成要素は、複数の Layer

からなる Dense block と、ダウン (アップ) サンプリングを行う Transition Down (Up) に分けることができる。構成要素の概要を表 1 に示す。

3.2.1 Layer

Dense block を構成する Layer は、バッチ正規化 (Batch Normalization)、活性化関数として ReLU (Rectified Linear Unit)、 3×3 convolution、そして dropout を組み合わせたものである。ここでの 3×3 convolution によって画像サイズは変化しない。Layer では、growth rate の数だけ新しい特徴マップが生成される。通常は 1 つの Dense blockにつき、Layer の処理は 4 から 15 回程度繰り返される。

3.2.2 Transition Down

Dense block によって出力された特徴マップの空間解像度を下げるために、Transition Down (TD) が導入される。TD は、バッチ正規化、ReLU、 1×1 convolution、dropout へと続き、最後にサイズが 2×2 の max pooling が行われる。pooling 操作によって、空間解像度は $1/4$ にダウンサンプリングされる。

3.2.3 Transition UP

Transition UP (TU) では入力画像の空間解像度を復元させるために、stride 2 の 3×3 transposed convolution が行われる。これにより、空間解像度は 4 倍にアップサンプリングされる。その後、特徴マップは pooling される前の同じ空間解像度を持つ特徴マップとショートカット接続を用いて連結され、次の Dense block の入力となる。

3.3 ネットワーク構造

このネットワークの構造は画像の特徴を抽出する Encoder と、特徴から目標画像を生成する Decoder から成る Encoder・Decoder モデルに従う。Encoder では、Dense block と TD が交互に使われる。Dense block で画像の特徴を取り出した後に、TD で空間解像度を徐々にダウンサンプリングする。Decoder では Dense block に加えて、TU とショートカット接続が行われ、画像がアップサンプリングされる。ここでの Dense block は、Encoder の時とは違い、入力の特徴を出力と連結させない。これは特徴マップ数の過剰な増加を防ぎ、演算処理にかかる負担を減らすためである。TU の後にショートカット接続が行われ、対応する特徴マップが連結され、高い空間解像度の情報を得る。ネットワークの最後の層である 1×1 Convolution により、目的の特徴マップ数を持つ画像が生成される。

提案手法における実験では、56 の畳み込み層から成る FC-DenseNet を用いた (図 3 参照)。Dense block 内の Layer 数は全て 4 つに固定し、growth rate は $k = 4$ に設定した。これにより、特徴マップは Dense block ごとに 16 個生成される。Dense block 間に TD と TU がそれぞれ対応して 4 回使われることで、RGB 画像の特徴を抽出し、元の画像サイズまで復元される。また、最後の Dense block に

表 1 Fully Convolutional DenseNets の構成要素
Table 1 Building blocks of Fully Convolutional DenseNets

Layer	Transition Down(TD)	Transition UP (TU)
Batch Normalization	Batch Normalization	
ReLU	ReLU	
3 × 3 Convolution	1 × 1 Convolution	3 × 3 Transposed Convolution
Dropout	Dropout	
	2 × 2 Max Pooling	

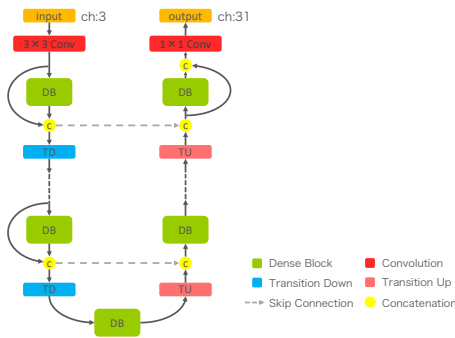


図 3 FC-DenseNet の構成
Fig. 3 FC-DenseNets architecture

限っては、入力の特徴を出力と連結させた。最終的に 1 × 1 Convolution の出力から 31 バンドのスペクトル画像を取得する。

4. 提案手法

4.1 スペクトルの表現

スペクトル画像を扱うにあたって、スペクトルの表現方法を定める必要がある。従来の深層学習を用いたスペクトル画像の超解像は、RGB 画像と出力スペクトル画像の関係をエンドツーエンド学習し、その際のスペクトルは離散的に表現される。本研究では、離散表現に加えて基底関数の和による表現も用いて実験を行った。基底関数表現を用いた学習では、ネットワークは離散表現された各バンドのスペクトル強度値を学習する代わりに、各基底関数の係数を学習する。基底関数には正規分布の確率密度関数と多項式を組み合わせたシグモイド関数の 2 種類を使用した。

4.1.1 離散

スペクトルを連続した値ではなく、離散的な値で表現することはシンプルな方法である。スペクトル画像の多くはカメラで対象物を撮影することで取得され、通常の RGB 画像よりもバンド数が多い画像である。その際、波長成分は多くの場合、一定の間隔ごとにサンプリングされる。カメラの技術的にもスペクトルを連続量として取得することは難しく、狭い間隔でサンプリングすることでスペクトルを近似している。したがって、スペクトル画像は、一般的にスペクトルを離散的に表現する。

また、画像の超解像は教師あり学習なので、出力形態はデータセットに依存する。学習に使うデータセットがスペ

クトル画像ならば、ネットワークは離散的に表現されたスペクトルを学習することになる。

先行研究では、RGB 画像の 3 バンドから学習用のスペクトル画像のバンド数までアップサンプリングした。深層学習を用いる従来手法では、データセットの入力 RGB 画像と出力スペクトル画像の関係をエンドツーエンドで直接学習している。

4.1.2 基底関数

主成分分析から、測定された代表的なスペクトルは少ない主成分で再現できることが知られている [17]。この考えに基づき、3 つの基底関数を足し合わせることで、スペクトルを再現する。

$$S(\lambda) = w_1 f_1(\lambda) + w_2 f_2(\lambda) + w_3 f_3(\lambda) \quad (4)$$

ここで、 λ は波長、 $S(\lambda)$ はスペクトル、 f_i は基底関数、 w_i は基底関数にかかる重みである。

また、自然界でのスペクトル分布は一般に連続性を持ち緩やかに変化するという特性 [18] を考慮して、基底関数には滑らかな曲線を持つ正規分布の確率密度と、jakob らが用いたシグモイド関数を用いて実験を行った。

基底関数の係数は画像の各ピクセルごとに求められ、ネットワークが出力するチャンネル数は係数の数と等しくなる。また、学習用スペクトル画像との誤差を評価するため、出力された係数からスペクトル画像を生成する必要がある。はじめに、出力された係数から基底関数に基づいて連続的な予測スペクトルが算出される。その後、学習画像のバンド数と等しくなるように予測スペクトルを一定間隔でサンプリングすることで、スペクトル画像が得られる。最後に画像間の誤差が計算され、最適な係数が学習される。

正規分布

スペクトルの基底関数として正規分布の確率密度関数は以下のように表される。

$$f_i(\lambda) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(\lambda - \lambda_{c,i})^2}{2\sigma_i^2}\right) \quad (5)$$

ここで、 $\lambda_{c,i}$ は波長の中心、 σ_i^2 は分散である。分散 σ_i^2 の値が大きいくほど、曲線はよりなだらかになる。

シグモイド

Jakob ら [7] は、シグモイド関数と多項式を組み合わせた関数を用いてスペクトルを表現した。本研究では、この関数を基底関数として以下のように基底関数の和

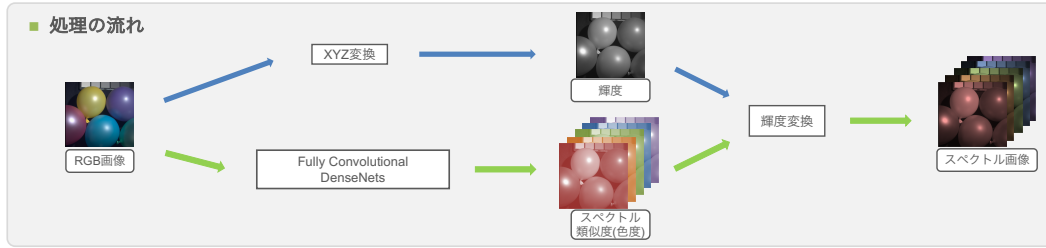


図 4 提案手法の処理の流れ

Fig. 4 Processing flow of the proposed method

による表現へと拡張した。

$$S(\lambda) = w_1 f_1(x) + w_2 f_2(x) + w_3 f_3(x) \quad (6)$$

$$f_i(x) = \frac{1}{2} + \frac{x}{2\sqrt{1+x^2}} \quad (7)$$

$$(x = c_{i,0}\lambda^2 + c_{i,1}\lambda + c_{i,2})$$

ここで、 $c_{i,j}$ は 2 次多項式の係数である。

4.2 RGB 画像からスペクトル画像への変換

提案手法はスペクトル類似度に基づくアプローチをとる。すなわち、スペクトル強度値を直接学習せず、代わりにスペクトルの類似度を学習する。学習したスペクトル類似度は後にスケールされ、元の RGB 画像の輝度と一致する。ここで、スペクトルの類似度と大きさは XYZ 表色系の xy 色度と輝度である Y 成分にそれぞれ相当する。

提案手法の処理の流れを図 4 に示す。また、提案手法の処理手順は以下の通りである。

- (1) 深層学習を用いて、RGB 画像から画像のスペクトル類似度を出力する。
- (2) RGB 画像と、出力されたスペクトル画像の輝度値を XYZ 表色系より計算する。
- (3) 計算された輝度値からスペクトルの輝度を変換する。

4.2.1 類似度を考慮した損失関数

n 個のバンド数を持つスペクトル画像では、1 ピクセルあたりのスペクトルは n 次元のベクトルとみなすことができる。2 つのベクトル \mathbf{a}, \mathbf{b} の類似度を測る一般的な方法として、次式に示すコサイン類似度がある。

$$\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|_2 \|\mathbf{b}\|_2} \quad (8)$$

ここで、 \cdot は内積、 $\|\cdot\|_2$ はユークリッドノルムである。コサイン類似度は 2 つのベクトルがどの程度同じ方向を向いているかを表し、区間 $[-1, 1]$ の値を取る。2 つのベクトルが同じ向きならばコサイン類似度は 1 となり、反対の向きならば -1 となる。

提案手法では 2 つのスペクトル $\lambda, \hat{\lambda}$ の類似度に関する誤差を計算する損失関数 $Loss$ を次式とする。

$$Loss(\lambda, \hat{\lambda}) = 1 - \cos(\lambda, \hat{\lambda}) \quad (9)$$

4.2.2 輝度値の取得

xy 色度には色相や彩度に関する情報は含まれているが、明るさに相当する輝度は含まれていない。xy 色度を用いて色を表す場合は、XYZ 表色系での輝度にあたる Y 成分が必要となる。スペクトル類似度から元の輝度幅のスペクトルへと再構成するためには、同様に Y 成分が必要となる。提案手法は RGB 画像から復元を行うので、輝度値 Y は RGB 値から取得できる。sRGB 表色系から Y は (10) 式により変換される。また、ネットワークから出力されたスペクトル $\hat{S}(\lambda)$ の輝度 \hat{Y} は、等色関数 \bar{y} を用いて (11) 式により計算される。

$$Y = 0.2126 \times R + 0.7152 \times G + 0.0722 \times B \quad (10)$$

$$\hat{Y} = \frac{1}{k} \int_{\Lambda} \hat{S}(\lambda) \bar{y}(\lambda) dx \quad \left(k = \int_{\Lambda} \bar{y}(\lambda) dx \right) \quad (11)$$

4.2.3 スペクトルの輝度変換

(10) 式と (11) 式により得られた輝度値を用いて、ネットワークから出力されたスペクトルを (12) 式によりスケールすることで輝度変換を行いスペクトル画像を得る。スケールングでは、スペクトルの色度は正確に保持される。

$$S(\lambda) = \frac{Y}{\hat{Y}} \hat{S}(\lambda) \quad (12)$$

5. 実験と結果

5.1 実験方法

実験では 2 つの異なるデータセット、CAVE dataset[19] と VISTA dataset を用いた。公開されているデータセットはスペクトル画像に加えて対応する RGB 画像も提供されているが、正確に定量的な評価を行うため、XYZ 表色系の等色関数に従って入力データとなる RGB 画像を生成し直した。CAVE は画像サイズ 512×512 ピクセルで実写画像 32 枚から成り、VISTA は画像サイズ 640×480 ピクセルで本実験用に CG レンダリングされた 112 枚から成る。両データセットとも 400nm から 700nm の波長範囲を 10nm 間隔でサンプリングしており、スペクトルのバンド数は 31 である。また、データセットの画像枚数が多くないので評価指標として 2-fold cross-validation を行った。出力されたスペクトル画像は 8bit の輝度幅に変換された後に評価され、評価指標には二乗平均平方根誤差 (RMSE) を用いた。

表 2 MSE と CosineE の RMSE 評価値 (CAVE dataset)

Table 2 RMSE values of MSE and CosineE (CAVE dataset)

	離散	正規分布	シグモイド
MSE (従来)	6.442	8.113	5.989
CosineE (提案)	4.998	7.185	5.567

CNN の実装には PyTorch を用いた。損失関数には PyTorch のライブラリから従来手法としてエンドツーエンド学習を行う `nn.MSELoss` (MSE), 提案手法には `nn.CosineEmbeddingLoss` (CosineE) を利用した。スペクトルの表現は離散表現 (discrete) と, 基底関数の和による表現として正規分布の確率密度関数 (nd) とシグモイド関数 (sig) を使い, 合わせて 3 種類の表現方法で実験を行った。その 3 種類それぞれに, MSE と CosineE の 2 つを用いて, スペクトル画像を再構成した結果を比較する。

5.2 VISTA dataset の作成

本実験ではスペクトル画像を生成するため, 光源のスペクトルとオブジェクトのスペクトル反射率を用いて CG スペクトラルレンダリングを行った。レンダリングでは物理ベースレンダリングの 1 つである, プログレッシブフォトンマッピング (PPM) [20] を用いた。PPM はフォトンマッピングの推定精度がメモリ容量に制限されてしまう問題を解決した手法である。400nm から 700nm を 10nm ごとに一様にサンプリングして, 1 時間かけてレンダリングを行うことで画像を生成した。

光源には D65 光源と白色 LED の 2 つを使用した。自然光である D65 光源は連続的な滑らかな分布を持つのに対し, 人工的な光である白色 LED は輝線を持ち, 険阻なスペクトル分布を持つ。また, 異なるオブジェクトを 8 つ用意し, それぞれのオブジェクトは固有の材質を持つようにした。材質には金属類や透過性の物質を主に選んだ。1 つのシーンには図 6 に示すように 2 つの異なるオブジェクトを左右に配置した。また, 壁には Macbeth Color Cheker[21] からスペクトル反射率を選択し, シーンごとに壁の色を変えた。最終的に 112 枚のスペクトル画像を生成した。

レンダリングしたスペクトル画像は幅広い輝度幅を持つので, トーンマッピングを行って HDR 画像から LDR 画像に変換する必要がある。今回の実験では Reinhard ら [22] の手法を使用した。この処理により, 局所的に明るい部分の輝度値を下げ, 暗い部分の輝度値を上げることができる。

5.3 実験結果

各データセットでの RMSE 評価値を表 2, 表 3 に示す。損失関数について比較すると, どちらのデータセットにおいても従来手法の MSE よりも提案手法の CosineE を用いた方が誤差が小さくなった。また, 全体的に基底関数を用いた表現よりも離散表現の方が誤差は小さい。

表 3 MSE と CosineE の RMSE 評価値 (VISTA dataset)

Table 3 RMSE values of MSE and CosineE (VISTA dataset)

	離散	正規分布	シグモイド
MSE (従来)	9.726	22.638	58.558
CosineE (提案)	9.141	12.752	13.816

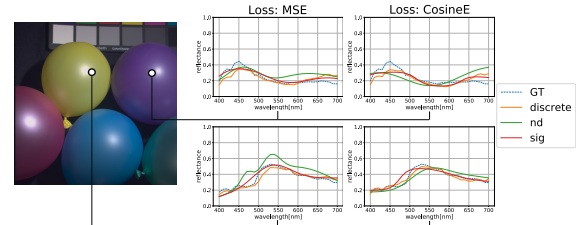


図 5 各手法でのスペクトル (CAVE dataset)

Fig. 5 Spectra for each method (CAVE dataset)

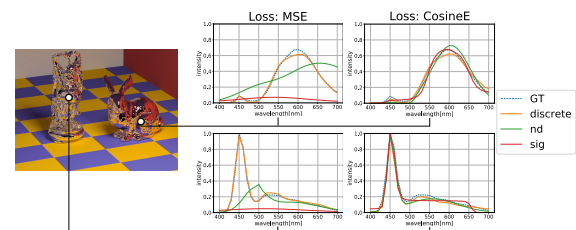


図 6 各手法でのスペクトル (VISTA dataset)

Fig. 6 Spectra for each method (VISTA dataset)

図 5 と図 6 では各データセットにおける出力スペクトルを損失関数別に表している。左列が MSE (従来手法), 右列が CosineE (提案手法) を用いた場合である。両図の右列のスペクトルから提案手法は輝度変換により, 元のスペクトルと同じ輝度幅に変換されていることが分かる。ほとんどの場合において離散表現が Ground Truth (GT) に近いスペクトルを出力している。

6. おわりに

本研究では, 従来のアプローチ (RGB 画像とスペクトル画像の関係性をエンドツーエンド学習) をとらず, スペクトル類似度を学習し, RGB 画像の輝度に基づいてスケールリングを行うことで RGB 画像からスペクトル画像に変換する新たな手法を開発した。提案手法は従来手法と比べて, スペクトル強度値との誤差が小さくなることを確認した。また, スペクトルの表現については今回の実験では, 3 つの基底関数の和による表現よりも, 離散表現の方が概ね結果が良いことが分かった。

今後の課題としては, 基底関数の最適な数の検討, 蛍光現象等のスペクトルや LDR 画像よりも幅広い輝度幅を持つ HDR 画像を用いた実験を行うことなどが挙げられる。

基底関数の数は今回は 3 つのみでしか実験していないが, 基底関数の数が多いほど任意のスペクトルを正確に表現できることを考えると, 基底関数の数を増やすことにより結

果が良くなる可能性がある。また、蛍光現象等の固有の分布を持つスペクトルは波長依存性が高い光学現象であり、それらのスペクトルを再現することも今後の課題として挙げられる。そして、今回実験で行った LDR 画像と異なり HDR 画像は実世界同様の幅広い輝度を持つので、LDR の RGB 画像から HDR のスペクトル画像への変換を実現することもチャレンジングな課題の一つである。

参考文献

- [1] Moroni, M., Mei, A., Leonardi, A., Lupo, E. and Marca, F. L.: PET and PVC Separation with Hyperspectral Imagery, *Sensors*, Vol. 15(1), pp. 2205–2227 (2015).
- [2] Belluco, E., Camuffo, M., Ferrani, S., Modenese, L., Silvestri, S. and Marani, A.: Mapping salt-marsh vegetation by multispectral and hyperspectral remote sensing, *Remote Sensing of Environment*, Vol. 105(1), pp. 54–67 (2006).
- [3] Ikeda, S., Watanabe, S., Raytchev, B., Tamaki, T. and Kaneda, K.: Spectral Rendering of Interference Phenomena Caused by Multilayer Films Under Global Illumination Environment, *ITE Transactions on Media Technology and Applications*, Vol. 3(1), pp. 76–84 (2015).
- [4] Yilmaz, O., Selimoglu, O., Turk, F. and Kirik, M. S.: SNR analysis of a spaceborne hyperspectral imager, *2013 6th International Conference on Recent Advances in Space Technologies (RAST), 2013 6th International Conference on*, IEEE (2013).
- [5] Dong, C., Loy, C. C., He, K. and Tang, X.: Image Super-Resolution Using Deep Convolutional Networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2015).
- [6] Otsu, H., Yamamoto, M. and Hachisuka, T.: Reproducing Spectral Reflectances From Tristimulus Colours, *Computer Graphics Forum*, Vol. 37(6), pp. 370–381 (2018).
- [7] Jakob, W. and Hanika, J.: A Low Dimensional Function Space for Efficient Spectral Upsampling, *Computer Graphics Forum*, Vol. 38(2), pp. 147–155 (2019).
- [8] Rei, K., Wright, J., Tai, Y. W., Ezra, M. B., Ikeuchi, K. and Matsushita, Y.: High resolution hyperspectral imaging via matrix factorization, *the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2329–2336 (2011).
- [9] Li, Y., Hu, J., Zhao, X., Xie, W. and Li, J.: Hyperspectral image super-resolution using deep convolutional neural network, *Neurocomputing*, Vol. 266, pp. 29–41 (2017).
- [10] Guadarrama, S., Dahl, R., Bieber, D., Norouzi, M., Shlens, J. and Murphy, K.: PixColor: Pixel Recursive Colorization, *British Machine Vision Conference* (2017).
- [11] Nguyen, M. H., Prasad, D. K. and Brown, M. S.: Training-Based Spectral Reconstruction from a Single RGB Image, *European Conference on Computer Vision*, pp. 186–201 (2014).
- [12] Arad, B. and ben Shahar, O.: Sparse Recovery of Hyperspectral Signal from Natural RGB Images, *European Conference on Computer Vision*, pp. 19–34 (2016).
- [13] Galliani, S., Lanaras, C., Marmanis, D., Baltsavias, E. and Schindler, K.: Learned spectral super-resolution, *arXiv preprint arXiv:1703.09470* (2017).
- [14] Jégou, S., Drozdal, M., Vázquez, D., Romero, A. and Bengio, Y.: The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation, *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2017).
- [15] Huang, G., Liu, Z., v.r Maaten, L. and Weinberger, K. Q.: Densely Connected Convolutional Networks Publisher: IEEE, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).
- [16] He, K., Zhang, X., Ren, S. and Sun, J.: Deep Residual Learning for Image Recognition, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).
- [17] Cohen, J.: Dependency of the spectral reflectance curves of the Munsell color chips, *Psychonomic Science*, Vol. 1(1), pp. 369–370 (1964).
- [18] Maloney, L. T.: Evaluation of linear models of surface spectral reflectance with small numbers of parameters, *Journal of the Optical Society of America*, Vol. 3(10), pp. 1673–1683 (1986).
- [19] Yasuma, F., Mitsunaga, T., Iso, D. and Nayar, S. K.: Generalized assorted pixel camera: postcapture control of resolution, dynamic range, and spectrum, *IEEE Trans Image Process.*, Vol. 19(9), pp. 2241–2253 (2010).
- [20] Hachisuka, T., Ogaki, S. and Jensen, H. W.: Progressive photon mapping, *ACM Trans. Graph.*, Vol. 27(5), pp. 1–8 (2008).
- [21] McCamy, C. S., Marcus, H. R. and Davidson, J. A.: A color-rendition chart, *J. Appl. Photogr. Eng.*, Vol. 2(3), pp. 95–99 (1976).
- [22] Reinhard, E., Starck, M., Shirley, P. and Ferwerda, J.: Photographic tone reproduction for digital images, *ACM Transactions on Graphics*, Vol. 21(3), pp. 267–276 (2002).