Check for updates

Deep learning models for optically characterizing 3D printers

DANWU CHEN[1,2,*] AND PHILIPP URBAN[1,3]

[1] *Fraunhofer Institute for Computer Graphics Research IGD, Darmstadt, Germany*
[2] *Department of Computer Science, Technical University of Darmstadt, Darmstadt, Germany*
[3] *Norwegian University of Science and Technology NTNU, Gjøvik, Norway*
*danwu.chen@igd.fraunhofer.de

Abstract: Multi-material 3D printers are able to create material arrangements possessing various optical properties. To reproduce these properties, an optical printer model that accurately predicts optical properties from the printer's control values (tonals) is crucial. We present two deep learning-based models and training strategies for optically characterizing 3D printers that achieve both high accuracy with a moderate number of required training samples. The first one is a Pure Deep Learning (PDL) model that is essentially a black-box without any physical ground and the second one is a Deep-Learning-Linearized Cellular Neugebauer (DLLCN) model that uses deep-learning to multidimensionally linearize the tonal-value-space of a cellular Neugebauer model. We test the models on two six-material polyjetting 3D printers to predict both reflectances and translucency. Results show that both models can achieve accuracies sufficient for most applications with much fewer training prints compared to a regular cellular Neugebauer model.

1. Introduction

An optical printer model is a predicting function from an arrangement or ratio of printing materials to the optical properties of the resulting print. *Optical characterization* of the printing system is the process of fitting the model's parameters, which requires test prints and optical measurements. For appearance 3D printing the model needs to be inverted to reproduce distinct optical properties or visual attributes, such as color or translucency. Therefore, the accuracy of these models plays a crucial role for the final print quality.

1.1. Material arrangements, material cross-contamination and dot gain

Material arrangements in multi-material printing systems are specified by a voxel distribution in print resolution. A voxel is the smallest geometric structure (cuboid) that a 3D printing system can address. Most systems use anisotropic voxels with a much smaller extent orthogonal to the slice plane. Commonly each voxel can be filled with only one material. Some printing systems allow variable drop sizes and can fill a voxel with multiple materials.

The printer model has to consider two major effects:

1. Mechanical dot gain caused by material cross-contamination: Due to mechanical limitations, a printing system is generally not capable of filling a voxel with the intended material(s) perfectly. There is some cross-contamination of materials between neighboring voxels. In 3D printing, a mixing of materials across voxels is also necessary to ensure the structural integrity of the object.

2. Optical dot gain caused by light transport blurring: For translucent printing materials, incident light is transported through multiple voxels before it leaves the objects. Its spectral power distribution (SPD) is altered by all voxels in its path. Voxels filled with dark material surrounded by voxels filled with bright materials appear therefore bigger, which is called *optical dot gain*.

Light transport is described by the Radiative Transfer Equation (RTE) [1] and requires the knowledge of the intrinsic optical properties of the printing materials, i.e. spectral absorption, scattering and phase function. Furthermore, refractive indexes of the printing materials are required since Fresnel reflection at the material-air interface and between the materials impact light transport as well. Using the RTE it is principally possible to simulate the light transport within heterogeneous media using a Monte-Carlo-based path tracer [2]. This was done in the context of 3D printing by Sumin *et al.* [3], who employed the physical-based rendering engine Mitsuba [4] for local path tracing. There are, however, various problems with Monte-Carlo-based approaches:

1. It is difficult to measure the intrinsic optical properties of the printing materials.

2. It is difficult to measure material cross-contamination (material mixing between voxels) but it must be considered for accurate simulations.

3. Due to the high resolution of modern 3D printing systems, Monte-Carlo approaches are slow. Even for computing the appearance of small objects (a few centimeters), billions of voxels must be considered and a simulation may take days on today's consumer hardware.

Because of the shortcomings of Monte-Carlo-based approaches, optical printer models use either phenomenological quantities such as measured reflectances of selected printing material arrangements or simplifications of the RTE.

Almost all models do not directly use voxel-based material arrangements as input but material ratios (or transformed ratios) instead. This assumes intrinsically that material distributions are dominated by high spatial frequencies minimizing material agglomerations. Modern 3D halftoning methods produce material arrangements possessing such *blue-noise characteristics* [5]. This minimizes graininess and apparent banding artifacts, particularly in smooth color transitions, by exploiting the human visual system's low contrast sensitivity for high spatial frequencies. We refer to material ratios or transformed material ratios as *tonal values* and denote the tonal value space by $\mathcal{T}$.

1.2.   Measuring optical properties

In this paper, we aim to accurately predict color and translucency of the arrangement of printing materials corresponding to each tonal value in $\mathcal{T}$. To fit an optical printer model and to evaluate its accuracy, we need a thorough specification of how color and translucency of test prints are measured:

For color, we measure reflectance spectra in an off-specular 45/0 measurement geometry [6] as shown in Fig. 1(a) using a white backing and compute CIELAB values for the CIE D50 illuminant and the CIE 1931 observer. For measuring the reflectances of highly translucent materials, edge loss [7] must be considered, which is a darkening effect caused by subsurface light transport away from the illuminated area. For this, the illuminated area D must be much bigger than the detection area A [8] (both is adjusted by circular apertures). To avoid measurement bias by edge loss for highly translucent printing materials we use $D \approx 90A$. The reflectance corresponding to a tonal value $v \in \mathcal{T}$ is determined by

$$r_{\lambda,D}(v) = r_\lambda^w \frac{\int_A E_{\lambda,D}(v,x)\,dx}{\int_A E_{\lambda,D}^w(x)\,dx} \tag{1}$$

where $\lambda \in [400\text{nm}, 700\text{nm}]$ is the wavelength, $E_{\lambda,D}(v,x)$ is the radiance emitted perpendicularly from the print's surface at point $x \in \mathbb{R}^2$ when illuminated at area D, $E_{\lambda,D}^w(x)$ is the radiance emitted perpendicularly from the surface of the opaque white reference (Lambertian) at point x using the same illumination and r_λ^w is the reference white's reflectance.
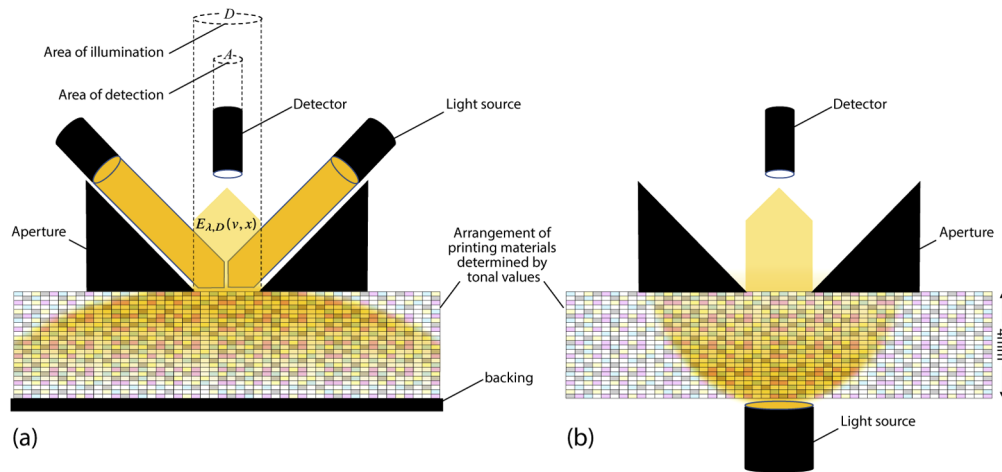
Fig. 1. Setup for reflectance (a) and transmittance (b) measurements for determining lateral and vertical light transport within printing material arrangements (shown here idealistically in a voxel grid) to compute the translucency quantity α [11]. Setup (a) with white backing and $D \approx 90A$ is used for spectral color measurements.

For translucency, optical properties can be light transport distances/profiles parallel to the surface normal [9,10]. In this paper, we use the one-dimensional quantity α proposed by Urban *et al.* [11]. It considers subsurface light transport parallel and orthogonal to the surface normal creating visual cues (shine through effects, reduction of high-spatial-frequency luminance contrasts) used by the human visual system to judge the degree of translucency [12,13]. For measuring α, two spectrophotometric reflectance measurements using black backing (see Fig. 1(a)) and a spectrophotometric transmittance measurement (see Fig. 1(b)) are required in addition to the color measurement. The two reflectance measurements use different illumination areas, D_1 and D_2, adjusted by two apertures. D_1 is equal to the detection area, i.e. $D_1 = A$, causing severe measurement bias by edge loss for highly translucent materials and D_2 is much bigger than the detection area, $D_2 = 16A$, resulting in less edge loss. The resulting reflectance measurements, $r_{\lambda,D_1}(v)$ and $r_{\lambda,D_2}(v)$, are used to compute the edge-loss difference [7], a quantity that describes subsurface light transport distances orthogonal to the surface normal. The definition of the α-value for a printing material arrangement defined by the tonal value $v \in \mathcal{T}$ is based on vertical light transport determined by the measured transmittance $t(v)$, lateral light transport determined by the edge loss difference measured by $r_{\lambda,D_1}(v)$ and $r_{\lambda,D_2}(v)$, and the lightness of the color measurement. We refer for more details to Urban *et al.* [11].

1.3. Resource efficiency and contribution

In addition to accuracy, an important property of optical printer models is resource efficiency, i.e. the number of required training prints and measurements to fit the model's parameters shall be as small as possible without scarifying accuracy. This is particularly important because multimaterial 3D printing is time-consuming and expensive, w.r.t. machine and material costs. Furthermore, optically characterizing a printing system is a recurring process because of inter- and intra-printer variability and the rapidly growing number of combinable printing materials.

In this paper we make the following contributions:

1. We propose a deep learning model for accurately predicting optical properties of a print from tonal values and a learning strategy requiring only a small number of training samples.

2. We propose a deep-learning-based approach for multidimensionally linearizing the cellular Neugebauer model to improve its prediction accuracy requiring only a small number of training samples.

3. We propose a loss function for training both models balancing reflectance, color and translucency errors using the ratio of the corresponding just noticeable differences under indoor viewing conditions.

2. Related work

2.1. Phenomenological models

Simple phenomenological optical printer models are adapted from 2D color printing. They usually operate on so-called *effective tonal* values computed from the input tonal values, called *nominal tonals*, to correct for mechanical dot gain. This correction is coined *linearization* and uses mostly 1D-functions per tonal [14] but also higher-dimensional functions have been proposed [15]. For duotone printers, e.g. using white and black materials, the simplest phenomenological model is the Murray-Davies model [16], which is just a linear interpolation between reflectance spectra of the printing materials. The canonical extension to N inks is called Neugebauer Model [17], that uses multilinear interpolation of the so-called Neugebauer primary reflectance spectra corrected for mechanical dot gain. The Neugebauer primaries consists of the base-materials and any mixture with equal material ratios, i.e. an N material printer has 2^N Neugebauer primaries. The interpolation weights can be determined by the Demichel equations [18,19]. The prediction performance of the Neugebauer model is rather low already in 2D printing [20] because it neglects subsurface light transport. This can be addressed by an extension introduced by Yule and Nielson [21–23] who applied a simple power function to the Neugebauer primaries. Various enhancements of the Yule-Nielson modified spectral Neugebauer model were proposed [24,25] yielding a prediction performance within the noise-level of 2D printing systems. For 3D printing systems, the Yule-Nielsen modified Neugebauer model, however, performs poorly [26]. Furthermore, it is conceptually not well suited to model 3D printing systems able to reproduce multiple levels of translucency, since using just one parameter (Yule-Nielson n-value) intrinsically relies on constant translucency of the substrate.

Another popular model is the Clapper-Yule model [27,28]. It considers multiple internal reflections at the material-air interface and can be adapted to various materials [29,30]. Also the Clapper-Yule model intrinsically assumes constant translucency of the substrate [31].

A way to arbitrarily improve the prediction performance of phenomenological models is to separate the tonal-value space into cells and evaluate the models within each cell. The most popular model is the cellular Yule-Nielsen spectral Neugebauer model and its modifications [15,32]. The improved model performance comes at the expense of more training prints and spectral measurements. For joint color and translucency reproduction with a 6-material 3D printer more than 3500 patches were necessary to achieve an average CIEDE2000 prediction error of 2.3 that is sufficient for most applications [33]. Babaei and Hersch proposed a cellular Yule-Nielsen modified spectral Neugebauer Model using barycentric subdivision of the tonal value space [34]. This reduces the number of required prints and measurements particularly if many materials are used. A disadvantage is that the barycentric subdivision is less localized than the common cube-based subdivision and the interpolation yield larger maximum errors.

2.2. Models based on RTE-simplifications

The advantage of the models based on RTE-simplifications is the limited number of necessary printed samples required for parameter fitting.

The simplest model is based on the Kubelka-Munk-Theory [35] that only considers two opposite collimated fluxes parallel to the surface normal. This two-flux approximation of the RTE

was used to model light propagation in layered arrangements of materials with different refractive indexes to fabricate spatially-varying translucencies [9,10]. The Kubelka-Munk-Theory assumes an infinite extent of the layers and their homogeneity. Since this does not hold for high-frequent 3D-halftones, the resulting color errors are rather high. Hensley and Ferwerda reported mean CIEDE2000 errors of 3.45 [26] which is unacceptable for most applications. To account for Fresnel reflection a Saunderson correction is often used [36].

A four-flux approximation of the RTE has been used for modeling 2.5D (relief) prints. In addition to the fluxes considered by the Kubelka Munk theory it considers also diffuse up and down facing fluxes. Van Song *et al.* reported maximal CIE94 error rates of less than 2 [37,38]. Simonot *et al.* developed a four-flux model relying on a matrix formalism and extended it to predict the bidirectional scattering distribution function of a stack of layered materials [39].

2.3. Neural-network-based models

Already more than a quarter of a century ago neural networks were used to invert optical printer models (also called separation) in 2D printing, i.e. computing the tonal values necessary to reproduce a given color [40] or to reproduce the same color as shown on a display [41]. Littlewood *et al.* [42] used a one-hidden-layer feedforward network to model the relationship between tonal values and color.

Shi *et al.* [43] concatenated a neural network of a backward spectral model (for separation) and a forward spectral model optimizing for both spectral RMSE and the CIE76 color difference formula for multiple illuminants. For the forward spectral model they used a fully-connected feed-forward neural network with 4 hidden layers, each with 300 neurons. They reported mean CIEDE2000 errors of the forward spectral model ranging between 1.5 and 2.5 but relatively large maximum errors of approx. 12 for the considered illuminants.

3. Precondition and problem formulation

We assume that the 3D printing system can be controlled by all values in the M-dimensional tonal space $\mathcal{T} = [0, 1]^M$, which is the input space of the proposed optical printer models. These values do not represent the material ratios whose sum must be one for 3D printing. This *unity condition* is also the reason why it is not necessary to consider a tonal value for the white printing material since we can compute it from the other tonals. For instance, for a printing system employing 5 printing materials CMYKW, the tonal value space $\mathcal{T}$ corresponds just to CMYK and $M = 4$. To ensure the unity condition, the values in $\mathcal{T}$ must be converted to material ratios by a transform $\mathbf{H} : \mathcal{T} \mapsto [0, 1]^M$, with $\|\mathbf{H}(t)\|_1 \leq 1, \forall t \in \mathcal{T}$ from which the fraction of white material can be computed as follows $t_w(t) = 1 - \|\mathbf{H}(t)\|_1$. $\mathbf{H}$ is part of the 3D printing pipeline before 3D halftoning (see Fig. 2) and can be decomposed into two subsequent transforms $\mathbf{H} = \mathbf{P} \circ \mathbf{L}$, where $\mathbf{P}(t) = (t_1/\max\{\|t\|_1, 1\}, \ldots, t_M/\max\{\|t\|_1, 1\})$ is a projection of the tonals to ensure the unity condition. $\mathbf{L} : [0, 1]^M \mapsto [0, 1]^M$ can be the identity but we recommend using an invertible transform similar to the transforms from nominal to effective tonals (e.g. 1D-per-tonal curves as described in [14]). This aids the selection of printing patches to fit/train the optical printer model because a uniform distribution of training values in $\mathcal{T}$ results in a more uniform distribution of optical quantities.

Fig. 2. Multi-material 3D printing pipeline.

We propose two deep learning-based approaches to predict optical material properties from tonal values: A *Pure Deep Learning* (PDL) model that is essentially a black-box without any physical ground and the *Deep-Learning-Linearized Cellular Neugebauer* (DLLCN) model that

uses deep-learning to multidimensionally compute effective tonals from nominal tonals. For the sake of simplicity, we just use two optical quantities that state-of-the-art 3D printing systems can both reproduce within physical limits to describe the approach: Spectral reflectance to predict color and α for translucency [11]. It is worth mentioning that both approaches are not relying on any special properties of these quantities except their boundedness. Therefore, we believe that they can be canonically extended to predict other optical quantities if appropriate loss functions are selected.

The optical printer models $\mathbf{O} : \mathcal{T} \mapsto \mathcal{S} \times \mathcal{A}$ investigated in this paper predict spectral reflectances $r \in \mathcal{S} = [0,1]^N$ and translucency $\alpha \in \mathcal{A} = [0,1]$ from tonal values $t \in \mathcal{T}$. Here N is the number of considered wavelengths that is often set to $N = 31$ assuming a uniform sampling of the visible wavelength range [400nm, 700nm] in 10nm steps.

4. Pure deep learning (PDL) model

Multi-path fully-connected neural network: The PDL model is a fully-connected neural network that, as shown in Fig. 3(a), has multiple paths each corresponding to one predicting task. Built upon the input layer (tonal features) is the "trunk" that consists of a number of hidden layers, and it splits into two "branches" corresponding to the two tasks (the reflectance-predicting task and the translucency-predicting task) each has its own hidden layers till its own output layer. The trunk is to learn generic representations shared across tasks, and each branch is to learn task-specific representations in order to make final predictions for the individual task. Specifically in our model, the trunk has 4 hidden layers each has 200, 1500, 1500 and 1500 neurons respectively. The reflectance branch has one hidden layer with 200 neurons and an output layer with N neurons corresponding to N-dimensional reflectance that is then converted to 3-dimensional CIELAB, and the translucency branch has one hidden layer with 30 neurons and an output layer with 1 neuron corresponding to the α value.
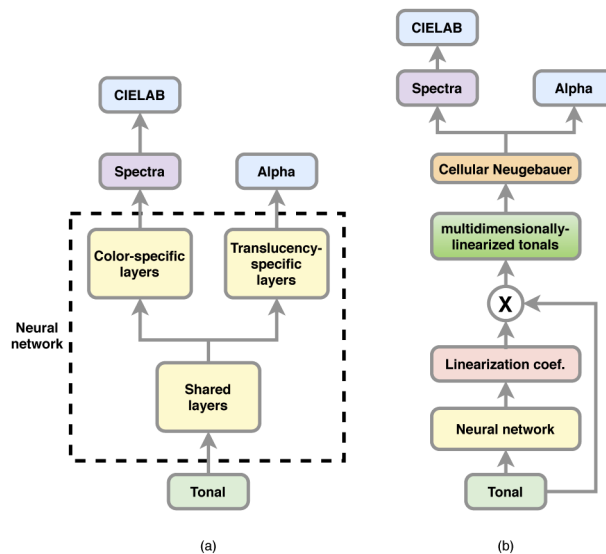

Fig. 3. The structure of the PDL (a) and DLLCN (b) models.

Activation functions to enable non-linearity capacity: For hidden layers, nonlinear activation functions are used in order to enable non-linearity capacity in the model. Specifically in our model, *leaky Rectified Linear Unit* (leaky-ReLU) activation function [44] is used. For the top layers of the 2 branches, a sigmoid activation function is used to ensure the predicted

values of reflectances and α are within $(0, 1)$. To alleviate the *saturation problem* of the sigmoid function $\sigma(x) = 1/(1 + \exp(-x))$ causing too tiny gradients during training [45,46], we altered it by $\widetilde{\sigma}(x) = \sigma(x + \sigma^{-1}(\mu_t))$, where μ_t is the training data's average for the corresponding quantity (reflectance value per wavelength, α). With this horizontally-shifted sigmoid, the neural network's output is closer to the training data, when initialized with small weights x as described, for instance, in [45]. This reduces initial loss and the chance of overshooting to the saturation regime of sigmoid.

5. Deep-learning-linearized cellular neugebauer (DLLCN) model

Figure 3(b) shows the structure of the DLLCN model. The DLLCN model is a composition of two functions

$$\mathbf{DLLCN} = \mathbf{CN} \circ \mathbf{DLL} : \mathcal{T} \mapsto \mathcal{S} \times \mathcal{A}. \tag{2}$$

The first function $\mathbf{DLL} : \mathcal{T} \mapsto \widetilde{\mathcal{T}}$ distorts the tonal value space aiming to multi-dimensionally linearize the domain of the second function, the cellular Neugebauer model $\mathbf{CN} : \widetilde{\mathcal{T}} \mapsto \mathcal{S} \times \mathcal{A}$ that predicts both reflectance and translucency. In this way, the degrees-of-freedom are reduced to a convex combination of Neugebauer cell-primaries ensuring plausible results. We use a deep neural network as the core of $\mathbf{DLL}$.

5.1. Structure of the linearization function

The core of the $\mathbf{DLL}$ is a fully-connected neural network $\mathbf{Z} : \mathcal{T} \mapsto \mathbb{R}^M$, where M is the dimensionality of the tonal value space $\mathcal{T}$. It has 5 hidden layers with 200, 400, 800, 400, and 200 neurons respectively and an output layer with M neurons. In order to avoid cross-contamination of tonals not present in the input, $\mathbf{Z}$ does not directly distort the tonal value space but computes weights of the input tonals $t \in \mathcal{T}$ defining $\mathbf{DLL}$ as follows:

$$\mathbf{DLL}(t) = \mathbf{C}_{[0,1]}[\mathbf{Z}(t) \odot t], \tag{3}$$

where $\odot$ is the Hadamard product (element-wise multiplication) and $\mathbf{C}_{[0,1]}$ is an operator that clips each entry to the range [0,1]. If, for instance, just two tonals corresponding to cyan and magenta materials are used in the input, the definition of $\mathbf{DLL}$ in Eq. (3) prevents the usage of any Neugebauer cell-primary with yellow or black material contribution.

Activation functions to enable non-linearity capacity: All hidden layers use leaky-ReLU activation function, and the output layer uses the linear activation function.

5.2. Cellular Neugebauer Interpolation

A traditional cellular spectral Neugebauer model is canonically extended to predict translucency by simply appending the translucency parameter $\alpha \in \mathcal{A}$ to its cell primary reflectance. In this way, the model can easily be extended to other optical quantities, since for any linearized tonal value the cellular Neugebauer model performs just a multi-linear interpolation of the cell primaries. For resource efficiency we recommend using just three grid points $\{0, 0.5, 1\}^M \subset \mathcal{T}$ for the cellular model.

6. Network design and learning strategy

6.1. Remarks on the neural networks design

The design of the neural networks for both models share the basic idea that the number of neurons should increase through the first layers to allow higher layers to have more capacity to capture high-level features of the tonal-optical relationship. Towards the output layer the number should decrease because the number of neurons of the output layer is small and the hidden layers should be able to summarize the features for making final predictions.

We selected the number of layers and total number of neurons so that the neural networks have enough capacity to learn even complex tonal-optical relationships. This design decision is critical because a too large neural network is prone to overfitting given the limited amount of training data. We selected these numbers to be most likely bigger than necessary to predict reflectance and translucency and use the strategies described in Sections 6.2 and 6.3 to avoid overfitting.

6.2. Batch normalization

Batch Normalization (BN) is used on each hidden layer to reduce Internal Covariate Shift (ICS) [47] and thus stabilize and accelerate the training. Batch Normalization helps tackle the vanishing/exploding gradients problem, which typically occurs for deep neural networks [45]. It also acts like a regularization technique, reducing generalization error [48].

6.3. Regularization to avoid overfitting

There are three major regularization techniques used in our approach:

1. Multitask learning [49], which allows samples to put more constraints on the part of a model that is shared across multiple tasks [48]. In our proposed models, as shown in Fig. 3, multitask learning is performed by simultaneously optimizing the reflectance-predicting task, the color-predicting task and the translucency-predicting task.

2. Dropout [50], as a computationally inexpensive but powerful regularization technique, is used on each layer in our models. The effectiveness of dropout comes from that it's approximately performing bagged ensemble on exponentially many thinned sub-networks that share parameters [48,50].

3. Early stopping, probably the most commonly used form of regularization technique in deep learning [48], is used in our model training. It considers a situation which is often observed in large models [48] that the training error decreases over time, but gradually the error on the validation set starts rising and the model starts overfitting. The training should be terminated when such a situation occurs, and the model parameters should return to the point where the validation loss is minimal. See Section 7.5 for more details of our experiment settings.

6.4. Loss function

The loss function serves two purposes:

1. It should ensure a high accuracy of predicting reflectances and translucency.

2. It should ensure a high color accuracy for specified viewing conditions, such as those used by the International Color Consortium (ICC) for defining the profile connection space [51]. These are the CIE D50 illuminant and the CIE 2° observer. For accurate color reproduction under selected viewing conditions this loss is very important because Euclidean distances in spectral space and perceived color differences are not well correlated (see e.g. [52]).

Therefore, our loss function has three parts:

1) Loss for reflectance-predicting task: For a predicted and measured reflectance $r_p, r_m \in \mathcal{S}$ the loss is computed using the Root-Mean-Square Error (RMSE):

$$\mathbf{E}_{\text{ref}}(r_p, r_m) = \sqrt{\frac{1}{N} \|r_p - r_m\|_2^2} \qquad (4)$$

2) Loss for color-predicting task: For above reflectances loss is computed by the CIEDE2000 formula [53]:

$$\mathbf{E}_{\text{col}}(r_p, r_m) = \Delta E_{00}(\mathbf{LAB}(r_p), \mathbf{LAB}(r_m)) \tag{5}$$

where $\mathbf{LAB} : [0, 1]^N \mapsto$ CIELAB is the function that computes CIELAB values from reflectances assuming specified viewing conditions. We use the CIE D50 illuminant and the CIE 2° observer.

3) Loss for translucency-predicting task: For a predicted and measured translucency $\alpha_p, \alpha_m \in \mathcal{A}$ the loss is computed as the absolute difference:

$$\mathbf{E}_{\text{tra}}(\alpha_p, \alpha_m) = |\alpha_p - \alpha_m|. \tag{6}$$

Note that α was designed to be nearly perceptually uniform for a set of reference materials with varying absorption and scattering coefficients and isotropic phase function under front/side lit conditions [11]. Brunton *et al.* used also absolute α-differences to optimize translucency in 3D printing [33].

The final loss function is a weighted average of the three loss functions:

$$\mathbf{E} = \mathbf{E}_{\text{col}} + a\mathbf{E}_{\text{ref}} + b\mathbf{E}_{\text{tra}} \tag{7}$$

where the weights $a, b \in \mathbb{R}$ are hyper-parameters that should be selected based on the relative magnitudes and importance of the loss functions. A color difference of CIEDE2000 = 1 is just very slightly above the *Just Noticeable Difference* (JND) and almost not detectable in a typical indoor lighting environment even if the colors are directly juxtaposed. A translucency difference of $\alpha = 0.1$ is also slightly above the JND for translucency [11]. Hence, to balance suprathreshold color and translucency differences very close to the JND in Eq. (7), we have to set $b = 10$. Tsutsumi *et al.* balanced CIEDE2000 and spectral RMSE linearly for spectral gamut mapping and found a value of $a = 50$ to be optimal [54]. Their results show however that for $a \in [20, 50]$ spectral and color accuracy is almost constant. An independent hyper-parameter tuning based on the validation data described in the experiments showed that $a = 20$ and $b = 10$ result in mean CIEDE2000 and α-errors close to JND with small spectral RMSE. We therefore recommend these values to achieve a good balance between color, spectral and translucency losses.

7.   Experiments and results

The purpose of the experiments is to evaluate the accuracy and resource efficiency of both models and to compare this with the traditional cellular Neugebauer (CN) model used, for instance, by Brunton *et al.* [33] to reproduce jointly color and translucency in 3D printing. Results and learning procedure can be reproduced as we show in Code 1 (Ref. [55]).

7.1.   3D printers and data sets

We have evaluated both models on two state-of-the-art material-jetting 3D-printers employing six materials: the Stratasys J750 and the Mimaki 3DUJ-553. Both printers use Cyan (C), Magenta (M), Yellow (Y), Black(K), White(W), Clear(Cl) build materials and one support material. The Mimaki 3DUJ-553 printer was equipped with the MH100 CMYKWCl materials, and the Stratasys J750 with Vivid CMY, Vero KCl and VeroPureWhite. The tonal value space $\mathcal{T}$ = CMYKG has five dimensions, where G $\in [0, 1]$ corresponds to the ratio of clear and white materials within the unit volume not occupied by the CMYK materials. The smaller G the larger is the fraction of white material. Very different halftoning algorithms were used for the two printers taking into account different material opacities, absorption levels, droplet sizes and droplet positioning capabilities of the machines, resulting in very different material arrangements and

optical properties for the same tonal values. It is worth mentioning that because of the high opacity and high absorption of the Mimaki prints the halftoning algorithm always adds some amount of clear material to avoid graininess and that this clear material has a slightly yellow tinge.

For the transformation **L** introduced in Section 3 we used a two-dimensional transformation per CMYK-tonal described in [33] and for G the identity.

Data was collected by printing flat targets with patches each corresponding to a special tonal value. Reflectances were measured in 45/0 geometry with a Barbieri Spectro LFP qb spectrophotometer equipped with a 19mm aperture of illumination and 2mm of detection to avoid edge loss for translucent materials. The translucency parameter α was measured with a Barbieri Spectro LFP according to [11]. The tonal values were selected for the initial purpose of fitting and investigating the performance of different cellular Neugebauer models/sub-models, which is the reason why the tonal value sets (reported here as 8-bit values) differ between printers.

Dataset for the Stratasys J750 We used a regular grid of tonal values $\{0, 64, 128, 191, 255\}^5 \subset$ CMYKG to print and measure 3125 patches in total.

Dataset for the Mimaki 3DUJ-553 The data was created by two Mimaki 3DUJ-553 printers with slightly different settings (firmware and post process) and will be therefore treated separately. We will refer to these printers as Mimaki 1 and Mimaki 2.

Mimaki 1: Samples were collected at 7 regular grid points in each of the CMYK dimensions for G = 0, and five regular grid points in each of the CMYK dimensions for G=245, 253, 254 and 255 respectively (see Table 1 for the grid point locations). This leads to $7^4 + 5^4 * 4 = 4901$ patches in total. Since the samples do not form a regular grid, the CN and DLLCN can't be used and only the performance of the PDL model is evaluated.

Table 1. Dataset for Mimaki 1 (grid point locations)

	G = 0	G = 245	G = 253	G = 254	G = 255
C	0,62,119,159,191,221,255	0,137,199,238,255	0,143,210,239,255	0,137,206,240,255	0,141,212,246,255
M	0,51,97,140,177,221,255	0,120,200,236,255	0,137,205,240,255	0,134,205,241,255	0,126,195,234,255
Y	0,90,115,128,148,171,255	0,172,224,244,255	0,184,225,244,255	0,184,236,250,255	0,161,224,250,255
K	0,75,142,170,203,230,255	0,200,237,249,255	0,193,241,252,255	0,190,241,253,255	0,155,236,252,255

Mimaki 2: Only opaque samples corresponding to G = 0 were collected. We used a regular grid $\{0, 43, 85, 128, 170, 213, 255\}^4 \subset$ CMYK of tonal values. Another 1099 random samples were also collected. Together, $7^4 + 1099 = 3500$ patches were printed.

7.2. Training and test data

We split for each printer the tonal value set $Q \subset \mathcal{T}$ of printed and measured samples (see Section 7.1) into two sets: The training/validation set $\mathcal{P}$ and the test set $\mathcal{E}$ with $\mathcal{P} \cap \mathcal{E} = \emptyset$. $\mathcal{P}$ contains always the set $\mathcal{P}_0$ of the necessary training samples of each model, i.e. the cellular Neugebauer primaries for the DLLCN (see Sec. 5.2) and the Neugebauer primaries for the PDL model. We first select 300 samples for $\mathcal{E} \subset Q \setminus \mathcal{P}_0$. For the Stratasys and Mimaki 1 printers $Q \setminus \mathcal{P}_0$ was split into 5 sets each corresponding to one of the five G-levels and 60 samples were selected randomly from each of these sets for $\mathcal{E}$. For the Mimaki 2 printer $\mathcal{E}$ was selected randomly from the 1099 random samples. Note that for all tonals in Q, printed and measured reflectances and α-values are available (*ground truth*).

In order to investigate the influence of the number of training samples on model performance, samples in $\mathcal{P}$ are increased starting with $\mathcal{P} = \mathcal{P}_0$ as follows: $\mathcal{P}_{i+1} := \{t_i\} \cup \mathcal{P}_i$, where the tonal value t_i is randomly selected from $Q \setminus \{\mathcal{E} \cup \mathcal{P}_i\}$. For the Mimaki 1 printer the new tonal value t_i is randomly selected from the set $\{(C, M, Y, K, G) \in Q \setminus \{\mathcal{E} \cup \mathcal{P}_i\} \mid G = G_{(i \bmod 5)}\}$. For the

Mimaki 2 printer, t_i is first randomly selected from the 1099 random samples not included in $\mathcal{E} \cup \mathcal{P}_i$. If all these 1099 samples are in $\mathcal{E} \cup \mathcal{P}_i$ it is selected randomly from $Q \setminus \{\mathcal{E} \cup \mathcal{P}_i\}$.

The traditional CN model requires a regular grid. To allow a comparison with a smaller amount of data, we used sub-grids as explained in the Supplement 1. For the non-aligned data of the Mimaki 1 printer the CN and DLLCN model cannot be evaluated. Table 2 summarizes the varying numbers of training samples for all three models.

Table 2. Number of samples used for training ($|Q|$, $|\mathcal{P}|$ is the cardinality of Q and $\mathcal{P}$)

Printer	$|Q|$	$|\mathcal{P}|$ for DLLCN and PDL	$|\mathcal{P}|$ for CN
Stratasys	3125	300, 400, 576, 1024, 1280, 1600, 2000, 2500 only PDL: 243, 350; only DLLCN: 500, 700	243, 576, 768, 1024, 1280, 1600, 2000, 2500
Mimaki 1	4901	200, 700, 1200, 1700, 2200, 2700	/
Mimaki 2	3500	500, 800, 1296, 1700, 2058, 2401 only PDL: 256; only DLLCN: 300	256, 500, 750, 1296, 2058, 2401

7.3. Computing and evaluating predictions

For training the DLLCN and PDL models, $\mathcal{P}$ is split into two sets, the training set $\mathcal{P}_t$ and the validation set $\mathcal{P}_v$, with $\mathcal{P} = \mathcal{P}_t \cup \mathcal{P}_v$ and $\mathcal{P}_0 \subset \mathcal{P}_t$. $\mathcal{P}_t$ is used for training the model and the validation set $\mathcal{P}_v$ is used for hyper-parameter training (e.g. for early stopping). $\mathcal{P}_v$ consists of 5% randomly selected samples from $\mathcal{P} \setminus \mathcal{P}_0$. The random splitting of $\mathcal{P}$ into $\mathcal{P}_t$ and $\mathcal{P}_v$ is performed 10 times resulting in 10 slightly differently trained models. Model predictions (reflectance, α) of the 10 models for $\mathcal{E}$ are averaged and used as the final prediction to evaluate the performance of the model trained on $\mathcal{P}$. Results of such averaged predictions are less biased by a single specific splitting of $\mathcal{P}$.

7.4. Software and hardware setup

Our model is implemented with TensorFlow 2.2.0, and is trained on an NVIDIA GeForce RTX 2080 SUPER GPU on a PC that has 64 GB RAM memory and an Intel Core i9-9900K CPU (3.6GHz). Training a model takes roughly 120s with 500 training samples, and roughly 290s with 2500 training samples. Predicting on 300 samples takes roughly 0.03s.

7.5. Training method

Xavier normal initializer [45] is used to initialize neural network weights. Dropout rate is set to 0.1 for PDL, and 0.05 for DLLCN. Adam optimizer [56] is used to train the models. The number of training epochs is 15,000. Batch size is equal to the size of the training set. The initial learning rate is 0.009. Learning rate decay is adopted, specifically, the learning rate is divided by a factor of 3 if the loss on validation set has not gone down for 1000 epochs. Early stopping is used during the training, specifically, the training is terminated if the loss on validation set has not gone down for 2000 epochs, and the model is rolled back to the point where the loss on validation set was minimal.

7.6. Results for the Stratasys printer

The benefit of using the machine learning-based models compared to the CN model can be seen by looking at accuracy vs. resource efficiency: To achieve a similar mean CIEDE2000 accuracy (see Fig. 4(a)) of the best result of the CN approach, which is CIEDE2000 = 2.4 corresponding to 2500 samples, the DLLCN approach only requires 500 samples which is a 5.0 times improvement in data efficiency, and the PDL approach further reduces the required number of samples to 300 which is an 8.3 times improvement. If all approaches use 2500 samples, the DLLCN achieves

CIEDE2000 = 0.9 and the PDL approach CIEDE2000 = 0.4 as the mean prediction error, which is a 2.7 and 6 times improvement in accuracy compared to the CN model. We see a similar trend for the 90th percentile (see Fig. 4(b)), which is CIEDE2000 = 4 for the CN model at 2500 training samples and almost 2.2 as big as for the DLLCN model that has CIEDE2000 = 1.8. The PDL approach achieves even CIEDE2000 = 0.7. To achieve a similar 90th percentile accuracy as the CN model for 2500 samples, the DLLCN model requires only 700 and the PDL model 350 samples, which is an approx. 3.6 and 7.1 times improvement.
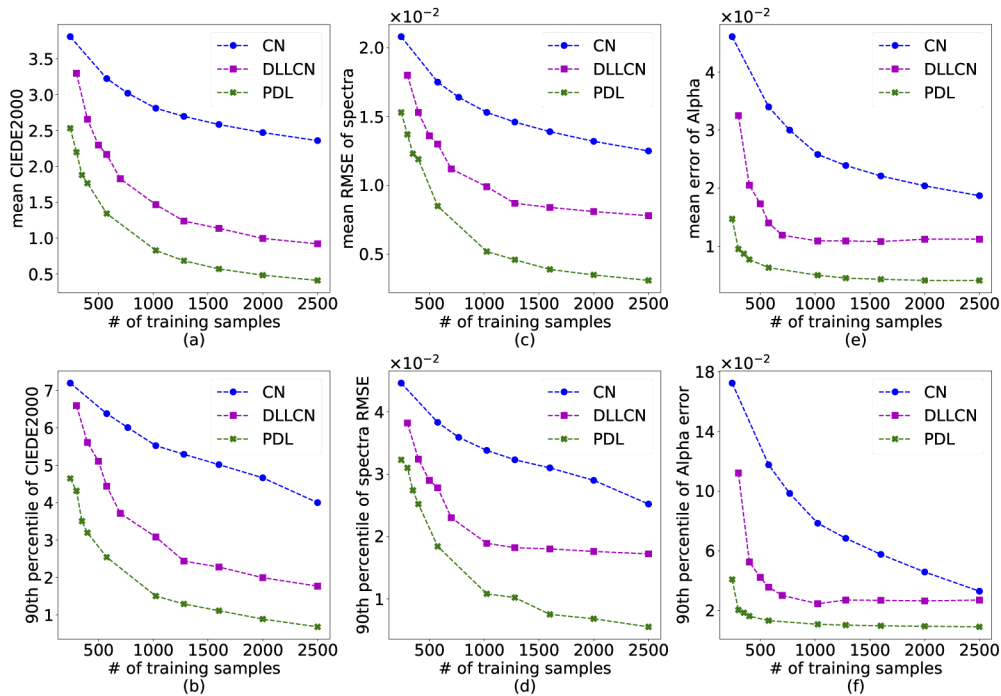
Fig. 4. Accuracy vs. dataset size plots for the Stratasys printer.

Mean and 90th percentile spectral RMSE as well as α errors follow the same trend as shown in Fig. 4(c)-(f). Note that the 90th percentile α-errors of the DLLCN and PDL model are below the just noticeable α-difference of approx. 0.1 for any considered training/validation set $\mathcal{P}$ (except for $|\mathcal{P}| = 300$ for DLLCN), indicating a sufficient prediction accuracy for nearly all applications. The CN model needs 768 training samples to pass this threshold.

7.7. Results for the Mimaki 1 printer

For the Mimaki 1 printer only the PDL model can be evaluated as described in Sec. 7.1. Due to the highly opaque white material and the resulting halftone method that uses almost always a fraction of clear material to avoid graininess, the relationship between tonals and optical properties of the resulting print is much more complex than for the Stratasys printer. Even though the accuracy vs. resource efficiency curves in Fig. 5 show the same trend as for the Stratasys printer, the absolute accuracy of the PDL model is worse. For the Mimaki 1 printer, 700 training samples are required to reach an average CIEDE2000 error of 2 and an 90th percentile error of 4. For the Stratasys printer this accuracy was reached with only 350 training samples. The α prediction is still below JND even for $|\mathcal{P}| = 200$, the smallest number of training samples.
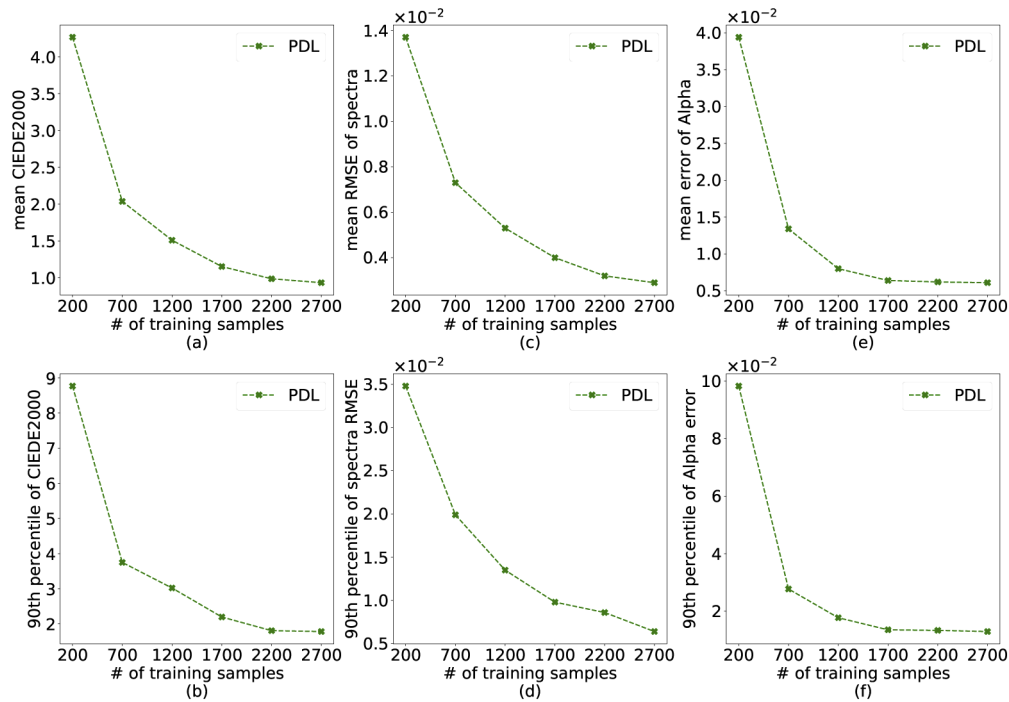
Fig. 5. Accuracy vs. dataset size plots for the Mimaki 1 printer.

7.8. Results for the Mimaki 2 printer

For the Mimaki 2 printer the same halftone method was used as for the Mimaki 1 printer, i.e. even though only opaque samples (G = 0) are considered the yellow-tinged clear material is used. Similar as for the other printers, we observe in Fig. 6 a diminishing returns of error rate reduction for the DLLCN and PDL model, i.e. in the beginning error rates drop strongly but the curve flattens for more training samples.

As shown in Fig. 6(a), the DLLCN model needs just 800 samples and the PDL model only 256 samples to reach a similar average color accuracy of CIEDE2000 = 1.9 as the CN model using 2401 samples. These are 3 and 9.4 times improvements in resources efficiency.

An interesting observation is that for the DLLCN model the spectral RMSE drops to a plateau and becomes even worse than for the CN model if more than 1296 training samples are used (See Fig. 6(c), (d)). This is because the degrees-of-freedom of the DLLCN model are limited to the spectral space spanned by the Neugebauer cell-primaries. Due to the contribution of the yellow-tinge clear material by the halftoning, some reflectances cannot be described by the linearized cellular Neugebauer model. We tested also a DLLCN model with four grid points (4GP) increasing the model's degrees-of-freedom to reproduce reflectances. It can be seen that even for the same number of samples the spectral RMSE is much smaller for the 4GP DLLCN model confirming the above statement. Interestingly, average color errors are not as much affected by the Neugebauer cell-primaries and both DLLCN models show similar performance (see Fig. 6(a)), indicating the benefit of optimizing models w.r.t. to both color and reflectances via multi-task learning. In contrast to the 4GP DLLCN model however, small improvements in average color accuracy of the 3GP DLLCN model for larger training data are at the expense of increased 90th percentile errors due to the model's limited degrees-of-freedom (see Fig. 6(b)).
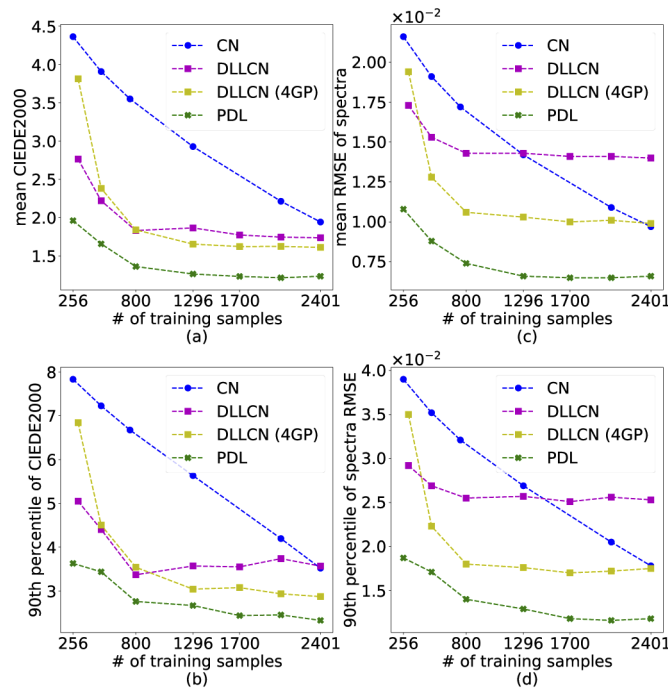
Fig. 6. Accuracy vs. dataset size plots for the Mimaki 2 printer.

7.9. *Visualization of spectral predictions*

In Fig. 7, we visualize the spectral predictions corresponding to the median and 99th percentile spectral RMSE error. For the latter case, the sRGB colors of its ground truth (GT) and its prediction are also shown. The figure shows that even using small training data the spectral predictions are quite consistent with the ground truths, and that the DLLCN and especially PDL models perform better than the CN model.
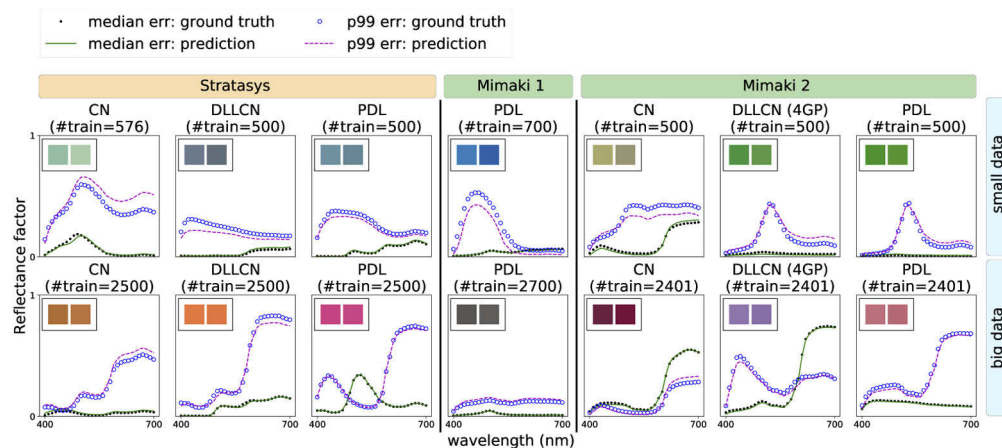

Fig. 7. Spectral predictions using different training data sizes.

8.   Conclusion and future work

For optically characterizing multi-material 3D printers, we proposed two deep learning approaches, the *Deep-Learning-Linearized Cellular Neugebauer* (DLLCN) approach and the *Pure Deep Learning* (PDL) approach, and presented a learning strategy to avoid overfitting. Both approaches outperform the standard cellular Neugebauer approach with large margins in terms of accuracy and data efficiency in predicting joint color and translucency validated on two state-of-the-art 6-material 3D printers. For the Stratasys J750 3D printer, they require up to 8.3/10.3 times fewer samples to achieve similar accuracy in color/translucency (α-value [11]) or are up to 6/4.6 times more accurate, when using the same amount of samples. For the Mimaki 3DUJ-553 printer, possessing a very complex tonal-optical relationship due to yellow-tinged clear material contribution in the 3D-halftoning, they need up to 9.4 times fewer samples to reach a similar color accuracy. Note that exactly the same model structures were used for both printers. Weights and hyperparameters were automatically determined by the described learning strategy.

The proposed deep-learning-based models possess a few intrinsic advantages compared to other phenomenological models. To achieve a sufficient accuracy in most applications they do not require large regular grids whose node number increases exponentially with the number of printing materials such as required by the cellular Neugebauer model. They can be updated with more scattered data, which is e.g. interesting for improving prints in particular optical regions of interest, e.g. colors and translucencies of prosthetic implants (teeth, eyes). Furthermore, they learn the impact of material cross-contamination and post-process treatment (polishing, coating) on the optical quantities of the resulting print. Both are difficult to measure and to consider by RTE-based models.

Future work will exploit similarities between 3D printing systems of the same type to further maximize resource efficiency of the optical characterization process. Even though such 3D printers possess intra- and inter-printer variability, they share low-level features of the tonal-optical relationship. To maximize resource efficiency, transfer learning [57] approaches shall be developed requiring less training data by exploiting fully trained models fitted to a different printer of the same type.

Disclosures. The authors declare no conflicts of interest.

See Supplement 1 for supporting content.

References

1. S. Chandrasekhar, *Radiative transfer* (Courier Dover Publications, 1960).
2. C. J. Zoller, A. Hohmann, F. Forschum, S. Geiger, M. Geiger, T. P. Ertl, and A. Kienle, "Parallelized monte carlo software to efficiently simulate the light propagation in arbitrarily shaped objects and aligned scattering media," J. Biomed. Opt. **23**(06), 1 (2018).
3. D. Sumin, T. Rittig, V. Babaei, T. Nindel, A. Wilkie, P. Didyk, B. Bickel, J. Kivának, K. Myszkowski, and T. Weyrich, "Geometry-aware scattering compensation for 3d printing," ACM Trans. Graph. **38**(4), 1–14 (2019).
4. W. Jakob, "Mitsuba renderer," (2010). http://www.mitsuba-renderer.org.
5. A. Brunton, C. A. Arikan, and P. Urban, "Pushing the limits of 3d color printing: Error diffusion with translucent materials," ACM Trans. Graph. **35**(1), 1–13 (2015).
6. ASTM, "ASTM E 1767: Standard Practice for Specifying the Geometry of Observations and Measurements for Characterizing the Appearance of Materials," (2011).
7. K. Yoshida, N. Komeda, N. Ojima, and K. Iwata, "Simple and effective method for measuring translucency using edge loss: optimization of measurement conditions and applications for skin," J. Biomed. Opt. **16**(11), 117003 (2011).
8. C. A. Arikan, A. Brunton, T. M. Tanksale, and P. Urban, "Color-managed 3d-printing with highly translucent printing materials," in *SPIE/IS&T Electronic Imaging Conference*, (San Francisco, 2015), pp. 9398.
9. Y. Dong, J. Wang, F. Pellacini, X. Tong, and B. Guo, "Fabricating spatially-varying subsurface scattering," ACM Trans. Graph. **29**(4), 1–10 (2010).

10. M. Hašan, M. Fuchs, W. Matusik, H. Pfister, and S. Rusinkiewicz, "Physical reproduction of materials with specified subsurface scattering," ACM Trans. Graph. **29**(4), 1–10 (2010).
11. P. Urban, T. M. Tanksale, A. Brunton, B. M. Vu, and S. Nakauchi, "Redefining A in RGBA: Towards a standard for graphical 3d printing," ACM Trans. Graph. **38**(3), 1–14 (2019).
12. R. W. Fleming and H. Bülthoff, "Low-level image cues in the perception of translucent materials," ACM Trans. Appl. Percept. **2**(3), 346–382 (2005).
13. I. Motoyoshi, "Highlight–shading relationship as a cue for the perception of translucent and transparent materials," J. vision **10**(9), 6 (2010).
14. D. R. Wyble and R. S. Berns, "A Critical Review of Spectral Models Applied to Binary Color Printing," Color Res. Appl. **25**(1), 4–19 (2000).
15. R. H. R. Rossier, "Introducing ink spreading within the cellular Yule-Nielsen modified Neugebauer model," in *IS&T/SID, 18th Color Imaging Conference*, (San Antonio, Texas, 2010), pp. 295–300.
16. A. Murray, "Monochrome reproduction in photoengraving," J. Franklin Inst. **221**(6), 721–744 (1936).
17. H. E. J. Neugebauer, "The Theoretical Basis of Multicolor Letterpress Printing (Translated D. Wyble and A. Kraushaar)," Color Res. Appl. **30**(5), 322–331 (2005).
18. E. Demichel, "Le procédé," **26**, 17–21 (1924).
19. E. Demichel, "Le procédé," **26**, 26–27 (1924).
20. R. Rolleston and R. Balasubramanian, "Accuracy of Various Types of Neugebauer Model," in *IS&T/SID*, (Scottsdale Ariz., 1993), pp. 32–36.
21. J. A. C. Yule and W. J. Nielsen, "The penetration of light into paper and its effect on halftone reproduction," in *Tech. Assn. Graphic Arts*, vol. 4 (1951), pp. 65–76.
22. J. A. C. Yule and R. S. Colt, "Colorimetric Investigations in Multicolor Printing," in *TAGA Proceedings*, (1951), pp. 77–82.
23. J. Viggiano, "Modeling the Color of Multi-color Halftones," in *TAGA Proceedings*, (1990), pp. 44–62.
24. R. Hersch, P. Emmel, F. Collaud, and F. Crété, "Spectral reflection and dot surface prediction models for color halftone prints," J. Electron. Imaging **14**(3), 033001 (2005).
25. R. Hersch and F. Crété, "Improving the yule-nielsen modified spectral neugebauer model by dot surface coverages depending on the ink superposition conditions," Proc. SPIE **5667**, 434–445 (2005).
26. B. D. Hensley and J. A. Ferwerda, "Colorimetric characterization of a 3d printer with a spectral model," in *Color and Imaging Conference*, (Society for Imaging Science and Technology, 2013), pp. 160–166.
27. F. Clapper and J. Yule, "Reproduction of color with halftone images," in *Proc. Seventh Ann. Tech. Meet. TAGA*, (1955), pp. 1–14.
28. F. Clapper and J. Yule, "The effect of multiple internal reflections on the densities of half-tone prints on paper," J. Opt. Soc. Am. **43**(7), 600–603 (1953).
29. R. Hersch, F. Collaud, and P. Emmel, "Reproducing color images with embedded metallic patterns," ACM Trans. Graph. **22**(3), 427–434 (2003).
30. G. Rogers, "A generalized clapper–yule model of halftone reflectance," Color Res. Appl. **25**(6), 402–407 (2000).
31. M. Hébert and R. D. Hersch, "Review of spectral reflectance models for halftone prints: principles, calibration, and prediction accuracy," Color Res. Appl. **40**(4), 383–397 (2015).
32. A. U. Agar and J. P. Allebach, "An iterative cellular ynsn method for color printer characterization," in *IS&T/SID*, (Scottsdale Ariz., 1998), pp. 197–200.
33. A. Brunton, C. A. Arikan, T. M. Tanksale, and P. Urban, "3d printing spatially varying color and translucency," ACM Trans. Graph. **37**(4), 1–13 (2018).
34. V. Babaei and R. D. Hersch, "*n*-ink printer characterization with barycentric subdivision," IEEE Trans. on Image Process. **25**(7), 3023–3031 (2016).
35. P. Kubelka and F. Munk, "Ein Beitrag zur Optik der Farbanstriche," Zeitschrift für Technische Physik **12**, 593–601 (1931).
36. J. Saunderson, "Calculation of the color of pigmented plastics," J. Opt. Soc. Am. **32**(12), 727–729 (1942).
37. T. P. Van Song, C. Andraud, and M. V. Ortiz Segovia, "Implementation of the four-flux model for spectral and color prediction of 2.5 d prints," in *NIP & Digital Fabrication Conference*, vol. 2016 (IS&T, 2016), pp. 26–30.
38. T. P. Van Song, C. Andraud, and M. V. Ortiz-Segovia, "Towards spectral prediction of 2.5 d prints for soft-proofing applications," in *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, (IEEE, 2016), pp. 1–6.
39. L. Simonot, R. D. Hersch, M. Hébert, and S. Mazauric, "Multilayer four-flux matrix model accounting for directional-diffuse light transfers," Appl. Opt. **55**(1), 27–37 (2016).
40. S. Tominaga, "Color control using neural networks and its application," in *Color Imaging: Device-Independent Color, Color Hard Copy, and Graphic Arts*, vol. 2658 (International Society for Optics and Photonics, 1996), pp. 253–260.
41. S. Abet and G. Marcu, "A neural network approach for rgb to ymck color conversion," in *TENCON'94-1994 IEEE Region 10's 9th Annual International Conference on:'Frontiers of Computer Technology'* (IEEE, 1994), pp. 6–9.
42. D. Littlewood, P. Drakopoulos, and G. Subbarayan, "Pareto-optimal formulations for cost versus colorimetric accuracy trade-offs in printer color management," ACM Trans. Graph. **21**(2), 132–175 (2002).
43. L. Shi, V. Babaei, C. Kim, M. Foshey, Y. Hu, P. Sitthi-Amorn, S. Rusinkiewicz, and W. Matusik, "Deep multispectral painting reproduction via multi-layer, custom-ink printing," ACM Trans. Graph. **37**(6), 1–15 (2019).

44. A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30 (2013), p. 3.

45. X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, (2010), pp. 249–256.

46. B. Xu, R. Huang, and M. Li, "Revise saturated activation functions," arXiv preprint arXiv:1602.05980 (2016).

47. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167 (2015).

48. I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1 (MIT press Cambridge, 2016).

49. R. Caruana, "Multitask learning," Machine learning **28**(1), 41–75 (1997).

50. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," The journal of machine learning research **15**, 1929–1958 (2014).

51. ICC , *File Format for Color Profiles* (International Color Consortium, http://www.color.org, 2010), 4th ed.

52. R. S. Berns, *Billmeyer and Saltzman's: Principles of Color Technology* (John Wiley & Sons, Inc., New York, 2000), 3rd ed.

53. CIE Publication No. 142, "Improvement to Industrial Colour-Difference Evaluation," Tech. rep., Central Bureau of the CIE, Vienna, Austria (2001).

54. S. Tsutsumi, M. R. Rosen, and R. S. Berns, "Spectral color management using interim connection spaces based on spectral decomposition," Color Res. Appl. **33**(4), 282–299 (2008).

55. D. Chen and P. Urban, "Python code for PDL and DLLCN models," figshare (2020), https://doi.org/10.6084/m9.figshare.13095743.

56. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980 (2014).

57. S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010).