A Gradient-Based Framework for 3D Print Appearance Optimization

THOMAS KLAUS NINDEL, Charles University, Czech Republic and Berufsakademie Sachsen, Germany TOMÁŠ ISER, TOBIAS RITTIG, and ALEXANDER WILKIE, Charles University, Czech Republic JAROSLAV KŘIVÁNEK, Charles University, Czech Republic and Chaos Czech a.s., Czech Republic



Fig. 1. Comparing optimization metrics for target (a) on a 2.5 mm slab. (b) was optimized for CIE dE76 using CMYKW materials. (c) adds a sixth orange material, improving color reproduction. (d) uses our combined metric with weights $\gamma_1 = 1$, $\gamma_2 = 9$, CMYKW. (e) shows the scale of all models used.

In full-color inkjet 3D printing, a key problem is determining the material configuration for the millions of voxels that a printed object is made of. The goal is a configuration that minimises the difference between desired target appearance and the result of the printing process. So far, the techniques used to find such a configuration have relied on domain-specific methods or heuristic optimization, which allowed only a limited level of control over the resulting appearance.

We propose to use differentiable volume rendering in a continuous materialmixture space, which leads to a framework that can be used as a general tool for optimising inkjet 3D printouts. We demonstrate the technical feasibility of this approach, and use it to attain fine control over the fabricated appearance, and high levels of faithfulness to the specified target.

CCS Concepts: • Computing methodologies \rightarrow Rendering; Volumetric models; • Applied computing \rightarrow Computer-aided manufacturing;

Additional Key Words and Phrases: Large-scale optimization, 3D Printing, volumetric light transport, volumetric optimization, computational fabrication, appearance reproduction, image metrics

Authors' addresses: Thomas Klaus Nindel, thomas@cgg.mff.cuni.cz, Charles University, KSVI MFF, Malostranské náměstí 25, Prague 1, 118 00, Czech Republic, Berufsakademie Sachsen, Hans-Grundig-Strasse 25, Dresden, 01307, Germany; Tomáš Iser, tomas@cgg. mff.cuni.cz; Tobias Rittig, tobias@cgg.mff.cuni.cz; Alexander Wilkie, wilkie@cgg.mff. cuni.cz, Charles University, KSVI MFF, Malostranské náměstí 25, Prague 1, 118 00, Czech Republic; Jaroslav Křivánek, Charles University, KSVI MFF, Malostranské náměstí 25, Prague 1, 118 00, Czech Republic, Chaos Czech a.s., Karlovo Námesti 288/17, Prague, 120 00, Czech Republic.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

@ 2021 Copyright held by the owner/author (s). Publication rights licensed to ACM. 0730-0301/2021/8-ART 178 15.00

https://doi.org/10.1145/3450626.3459844

ACM Reference Format:

Thomas Klaus Nindel, Tomáš Iser, Tobias Rittig, Alexander Wilkie, and Jaroslav Křivánek. 2021. A Gradient-Based Framework for 3D Print Appearance Optimization. *ACM Trans. Graph.* 40, 4, Article 178 (August 2021), 15 pages. https://doi.org/10.1145/3450626.3459844

1 INTRODUCTION

Resin-based, inkjet 3D printers create colored objects by carefully jetting microscopic droplets of base inks (e.g., CMYKW, cyan, magenta, yellow, black, and white) to form the object's volume. Due to the inks' translucency, the printouts exhibit complex volumetric light transport. On one hand, this is required for spatial color mixing and curing with UV light, but also leads to edge blurring and color-bleeding. This significantly complicates the reproduction of a desired appearance, i.e., finding the volumetric ink arrangement that leads to the best possible printout fidelity. Early approaches based on naïvely placing colored inks on the surface of an object, or using extrusion-based techniques, lead to blur, reduced contrast, and poor color reproduction.

Recently, new approaches have been presented to tackle these issues, and aim at closing the gap between the intended appearance and the printed result. These approaches solve different aspects of the problem: high color reproduction fidelity [Babaei et al. 2017; Brunton et al. 2015; Shi et al. 2018], alpha channel translucency printing [Brunton et al. 2018; Urban et al. 2019], or compensating for lateral scattering to increase texture sharpness [Elek et al. 2017; Sumin et al. 2019]. The latter techniques employ highly-accurate Monte Carlo light transport simulation inside the printouts, in conjunction with a heuristic-based operator to form an iterative optimization method.

However, all these methods are inherently local: They only consider a small neighborhood on the surface at a time, and do not fully account for the potentially global influence each voxel has on the objects appearance.

We formulate the problem of finding a suitable material arrangement as an *inverse volume rendering problem*. Given the appearance of a volume under specified viewing conditions, inverse volume rendering tries to recover its physical properties, so that a forward rendering pass would closely match the original views. Previous applications of inverse volume rendering have already lead to impressive results. They include reconstructing a voxel model of a smoke plume from still images [Gkioulekas et al. 2016], obtaining volumetric models of planetary nebulae [Magnor et al. 2004], or recovering spatially-resolved scattering and absorption properties of living tissue [Hochuli et al. 2016].

Using this approach in the setting of inkjet 3D printing, the entire optimization problem can be formulated under a common, gradient-based framework. Previously discussed use-cases follow naturally from this, lifting some of the inherent restrictions the existing forward-only, or purely heuristic approaches inhibit. For example, our method can reproduce translucent appearances while also matching previous scattering-compensation work in quality.

Leveraging a modern GPU renderer, our method runs on desktopgrade hardware with reasonable timings and can thus also be used for meta-optimizations such as per-target ink selection from an extended set of printer materials. Our contributions for optimizing the appearance of full-color inkjet 3D printouts are:

First, we present a flexible and robust framework based on numerical minimization. The method captures the global interdependence between changes in the voxel configuration and their result on the target appearance. It is general with respect to the optimization goal, the optical properties of available inks, and can handle arbitrary geometries. Our model is physically based, incorporating all parameters of volumetric light transport.

Second, we propose using a volume parameterization that implicitly confines the optimization to the physical limits of inks and the model's manufacturability, without the need for precomputed color tables. This allows the runtime to be (almost) independent of ink count, in turn opening up applications such as ink-set selection.

Third we design a combination of 3D error metrics that can express a range of visual stimuli to optimize for, while allowing for intuitive control over the process.

2 BACKGROUND AND RELATED WORK

2.1 Inverse volumetric light transport

Implementations of inverse light transport commonly employ a gradient-based optimization algorithm. The required gradients can be computed from finite differences or obtained by differentiable rendering, with Gkioulekas et al. [2013] being the first to suggest Monte Carlo estimates of derivatives for inverse rendering. Li et al. [2018] presented a differentiable renderer that allows the calculation of derivatives with respect to arbitrary scene parameters. In Zhang et al. [2019], a general, operator-based theory of radiative transfer was presented, along with an unbiased estimator for calculating derivatives with respect to arbitrary scene parameters, adding those

of volumetric light transport. Nimier-David et al. [2019] showed a retargettable renderer that can be configured to estimate the derivatives of light transport. Nimier-David et al. [2020] also recently proposed an adjoint method for back-propagating gradients, which greatly improves performance in settings with a large number of differentiable parameters. Since our approach requires calculating derivatives with respect to millions of volume parameters, this socalled "radiative backpropagation" technique is essential for us.

In Che et al. [2020], a learning-based technique is proposed that can be used for homogeneous inverse scattering applications; one of their networks was trained using a differentiable renderer, MitsubaDR, that was constructed by auto-differentiating a physically based renderer. Finally, Hašan et al. [2010] optimize material assignments for specified subsurface scattering properties using *discrete* optimization.

Khungurn et al. [2015] match the appearance of fibrous materials to several photographs using an inverse volumetric rendering pipeline. Papas et al. [2013] use a database of the scattering properties of pigment emulsions to iteratively find an optimal mixture that most closely matches a target appearance, which directly leads to a fabricable recipe.

2.2 Full-color fabrication

Color and beyond. While various publications tackled fabricating spatially-varying optical properties using 3D printing, e.g., subsurface scattering [Dong et al. 2010; Hašan et al. 2010], the first to show faithful *full-color reproduction* in inkjet 3D printing were Brunton et al. [2015]. They propose a geometry-adaptive error diffusion approach based on existing algorithms from 2D printing. Mapping 2D error diffusion filters onto the object's surface, they halftone the target layer by layer. They also discuss printer color management and building a color profile to allow gamut mapping.

Later, Brunton et al. [2018] extended their method to also support spatially varying translucency. Using lookup tables to map RGBA target colors to printer tonal values, they probabilistically insert transparent voxels into the volume. Translucency fabrication was further formalized by Urban et al. [2019], who provide a physically and perceptually meaningful definition of the alpha channel to create a device-independent standard for 3D printing translucency. Work was also done in controlling the surface reflectance of 3Dprintouts, either by directly printing microgeometry [Luongo et al. 2020; Rouiller et al. 2013], or based on a controlled application of varnishes [Piovarči et al. 2020]. Recently, Zheng et al. [2020] used neural networks and end-to-end optimization to create static 4D light fields that can then be fabricated with inkjet 3D printers.

Halftoning alternatives. Error diffusion halftoning is not the only method to obtain a discretized volume for printing. Arguing that there are inherent limitations of halftoning, Babaei et al. [2017] propose a solution they call *color contoning*. In contrast to halftoning, which works on the object's surface, their method drives "stacks" of inks below the object's surface. Doing this substantially removes halftoning artefacts.

Shi et al. [2018] take this approach even further. They use a neuralnetwork model for predicting color spectra of an ink stack. Their model can also be used in reverse, so that ink stacks, composed from their extended material library of 10 custom inks, can be used to achieve significantly improved color reproduction.

Full-color fabrication with Monte Carlo predictions. All the previously mentioned works are based on local color mixing, ignoring the contribution of global light transport. Elek et al. [2017] showed that by measuring the optical properties of basic inks, they can then use a full Monte Carlo simulation to faithfully predict the appearance of a 3D printout given its internal structure. These predictions are used as feedback in an optimization loop, heuristically pushing inks into the depth of the volume. Lateral scattering and color bleeding are reduced, leading to higher texture sharpness.

The original approach was limited to 2.5D planar slabs, but was later extended to full 3D geometry by Sumin et al. [2019]. They pay special attention to thin object geometries, where the colors of opposite faces create cross-talk. By computing a lookup table of color pairs achievable on opposing sides of variable-thickness slabs, they can perform content-aware gamut mapping. Their results show significant improvements of contrast and saturation for arbitrary object geometries.

Our method is inspired by Elek et al. [2017] and Sumin et al. [2019]. In contrast to their heuristic optimization, we employ a gradient-based optimization algorithm. By using end-to-end gradient estimates, our proposed method can determine to what degree every single voxel of the volume impacts the surface appearance, which is illustrated in Fig. 2, and improve the solution to lower the appearance error.

2.3 Image quality metrics

Numerical optimization methods are driven by quality metrics that compare intermediate results to a given target. We also require such metrics to be differentiable in order to be applicable within gradientbased optimization. Measuring the difference in appearance is a task similar to assessing image quality. This is a well-studied, yet challenging topic, mainly due to the non-linearity and subjectivity of the human visual system.

Beyond trivial measures like the mean squared error (MSE), *perceptual* quality metrics were designed to consider the human visual system. Color fidelity can, for example, be measured as the Euclidean distance in the perceptually uniform CIELAB color space (CIE76, McLaren [1976]), which was later expanded to CIEDE2000 [Luo et al. 2001]. Discontinuities in the latter prevent its usage in gradient optimization [Sharma et al. 2005].

Wang et al. [2004] proposed a structural similarity metric (SSIM) that also considers contrast and structure. The metric can be efficiently implemented using differentiable convolutions [Orihuela and Ebrahimi 2019], making it a good candidate for the use in gradient optimization and neural networks [Zhao et al. 2016a].

Due to its low sensitivity to a uniform bias, Zhao et al. proposed to combine SSIM with a color metric for an improved color accuracy. Chen et al. [2006] also noticed that SSIM favors blurred images over noisy ones, and suggested a modified edge similarity (ESSIM) metric. There are other SSIM alternatives available [Wang et al. 2003; Xue et al. 2013; Zhang et al. 2011], further improving its behavior. Notably, Preiss et al. [2014] propose a metric that is sensitive to differences in lightness, color and structure. Recent work [Zhang et al. 2018] has shown that deep neural networks originally trained for various computer vision tasks can also be used for image quality judgement. Their performance has shown to be favorable over metrics such as the previously disussed ones, when comparing to the quality assessments of the human observer.

3 PROBLEM STATEMENT

Overview. The method takes a textured 3D object on its input. The object has a certain *target appearance*, which is usually a diffuse color texture that defines how the object should look like when printed. This object is geometrically discretized into a voxel grid, where each *surface* voxel (texel) has a given target color based on the original texture. The goal is now to find a configuration of the 3D printer's inks that, when printed, would give a 3D printout that resembles the input as well as possible. In order to do that, an optimization loop is used that predicts how a candidate 3D ink arrangement would look like, and then compares it to the target appearance. An error is calculated using image quality metrics, and is backpropagated through to the ink arrangement via gradients, allowing its iteratively refinement. After convergence of the optimization, the results are quantized into an printable volume.

Formalization. We will now formalize the problem using the notation detailed in Table 1 such that it is solvable by numerical, gradientdescent optimization.

Given the geometrical discretization of a textured object as an input, we call the voxels on the object's surface "texels.¹" The set of all texels forms the target appearance, $T = (t_1, ..., t_I)$. Taking a given configuration of voxels *V*, we can predict the appearance *C* of the surface as

$$C = \Re(V), \tag{1}$$

where \Re represents the *volume rendering function*. Our goal is to find a configuration of voxels V_{opt} that minimizes a loss function e = E(C, T):

$$V_{opt} = \underset{V}{\arg\min E(C, T)}.$$
 (2)

To find V_{opt} using gradient-descent optimization, we require the gradient of the loss function w.r.t. the volume's parameters *V*. We can obtain the gradient by applying the chain rule:

$$\frac{\partial e}{\partial V} = \frac{\partial e}{\partial C} \frac{\partial C}{\partial V}.$$
 (3)

As *C* is a vector of the appearance of all texels c_i ,

$$\frac{\partial e}{\partial C} = E'(C,T) = \left(\frac{\partial e}{\partial c_i}\right)_{i=1}^{I}$$
(4)

represents the gradient of the loss function with respect to the appearance C.

Using then, for example, the sum of squared differences as the loss function:

$$E(C,T) = \sum_{i=1}^{l} (c_i - t_i)^2,$$
(5)

¹Mind that these are not *directly* connected to the fragments of the object's texture maps, but are volumetric elements obtained after UV mapping and resampling.

ACM Trans. Graph., Vol. 40, No. 4, Article 178. Publication date: August 2021.

Table 1. Mathematical Notation

Symbol	Usage
I, J, N	number of voxels at the volume's surface / total voxels
	of the volume / available printing materials
i, j, n	indices for surface voxels / volume voxels / materials
α_j	single scattering albedo of voxel <i>j</i> , $\alpha = \frac{\mu_s}{\mu_t}$
$\mu_{t,j}$	extinction coefficient of voxel j , $\mu_t = \mu_a + \mu_s$
$\mu_{s,n}$	scattering coefficient of material <i>n</i>
$\mu_{a,n}$	absorption coefficient of material n
ϕ	phase function
$\rho_{j,n}$	portion of the n -th material in the mixture of the j -th
	voxel
π	any volume parameter, e.g. $\mu_{t,1254}$ or $\rho_{421,2}$
S	space of the voxel parameters, formed by (μ_t, α, ϕ)
\mathbb{M}	space of the voxel parameters, formed by the material
	mixture ρ_n
M_n	parameters of printing material n , $M_n = (\mu_{a,n}, \mu_{s,n}, \phi)$
ci	appearance of the <i>i</i> -th surface voxel
С	appearance of the current solution, $C = (c_i)$
t _i	target appearance of the <i>i</i> -th surface voxel
Т	target appearance, $T = (t_i)$
V	volume parameters in space \mathbb{S} , $V = (\alpha_j, \mu_{t,j}, \phi_j)$
V'	volume parameters in material space \mathbb{M} , $V' = (\rho_{j,n})$
$\Re(V)$	volume rendering function, $C = \Re(V)$
z(V')	mixing function $V = z(V')$
е	error value
E(C,T)	loss function, $e = E(C, T)$



Fig. 2. Illustrating how voxels inside an object influence the color visible on the surface. A 2D slice through the profile of Nefertiti shows the singlescattering albedo in RGB per voxel. The inset visualizes the magnitude of Monte Carlo estimated $\partial c_i / \partial \alpha_j$ in one channel for the marked surface texel, i.e., how much each voxel influences the surface color in normal direction.

its gradient, consisting of the partial derivatives of *E* with respect to all volume parameters (here called π) can be obtained with:

$$\begin{pmatrix} \frac{\partial e}{\partial \pi} \end{pmatrix} = \sum_{i=1}^{I} \left(\frac{\partial e}{\partial c_i} \frac{\partial c_i}{\partial \pi} \right)$$

$$= 2 \sum_{i=1}^{I} \left((c_i - t_i) \frac{\partial c_i}{\partial \pi} \right), \forall \pi \in V$$

$$(6)$$

ACM Trans. Graph., Vol. 40, No. 4, Article 178. Publication date: August 2021.

Derivatives of the form $\partial c_i / \partial \pi$ encapsulate how the change in the appearance of one texel depends on the change of the parameter π . In other words, it is the first derivative of the rendering function with respect to this parameter. A visualization of these derivatives can be seen in Fig. 2.

By choosing an appropriate error function, the optimization is general with respect to the target appearance specification. It allows us to control whether the optimizer should emphasize, for example, texture sharpness, or optimize for highest possible color fidelity. We discuss this trade-off further in Sec. 4.5.

In order to embed this optimization task into the setting of inkjet 3D printing, a few extra steps are required: The discretization of the optimization results in a printable specification, which is discussed in Sec. 4.7, the voxelization of the objects geometry as an input to the pipeline, which can be obtained using distance fields (domain-specific discussions by Brunton et al. [2015]; Sumin et al. [2019]), and the optical properties of a set of printing materials can be obtained by various measurement setups, detailed in the state of the art report [Frisvad et al. 2020].

4 FORMING THE OPTIMIZATION LOOP

Using the relationship between the rendering function, the loss function and their derivatives as a starting point, we will now describe how to form a closed loop for the optimization of the volume configuration. Fig. 3 describes this loop graphically.

4.1 Derivative estimation

In practice, partial derivatives of the form $\partial c_i / \partial \pi$ can be obtained using a Monte Carlo estimator. Mathematically, it can be constructed from the path-space integral of light transport, as shown by Khungurn et al. [2015] and Zhang et al. [2019]. Implementations of these estimators usually work in what is called forward-mode: Parallel to computing radiance values, these algorithms also estimate the radiance derivatives alongside. Forward-mode refers to the order in which the graph of computations created by the radiance estimator is traversed for obtaining derivatives, in the context of rendering from scene parameters to radiance estimate. Forwardmode derivatives can be obtained automatically by a process called auto-differentiation (AD). We kindly refer the reader to Baydin et al. [2017] for an overview. This way of implementing derivatives requires one graph traversal for each differentiable parameter. Since we want to optimize millions of volume parameters, the associated computational cost quickly becomes prohibitive. Our initial experiments were based on such an estimator, taking tens of thousands of core-hours to optimize a 15 mm tall volume. Additionally, the resulting Jacobian matrix (in our case, derivatives of all texel values with respect to all volume parameters, a $I \times 2J$ matrix) leads to an unfavorable space-complexity. The use of reverse-mode AD, as implemented in Mitsuba2 [Nimier-David et al. 2019]), can ease this space-complexity by backpropagating derivative values through the renderer. However, reverse-mode AD requires storing computational graphs, which in case of a renderer can get very complex.



Fig. 3. Flow of data and their derivatives through the optimization loop. The voxelization of object geometry and the quantization of the optimization results are omitted for clarity. These are only executed once at the beginning and the end of the optimization process, respectively.

4.2 Radiative Backpropagation

We implement the volume derivatives using radiative backpropagation [Nimier-David et al. 2020]. It is a domain-specific instance of reverse-mode AD that eliminates the need for storing and traversing the renderers computational graph. Instead, the method propagates "derivative radiation" through the scene using light transport algorithms, an adjoint technique also seen in [Auzinger et al. 2018]. This radiation is emitted by the sensor and "consumed" by the scene parameters, effectively propagating all derivatives in one pass. This leads to major improvements in speed of computation and mitigates the storage requirements of traditional reverse-mode AD.

In our setting, the input for Radiative Backpropagation is derivative values $\partial e/\partial C$. As we are interested in derivatives with respect to the parameters of volumetric light transport, the output is $\partial e/\partial V$ (see Eq. 3, the second factor).

4.3 Derivatives in material space

Assuming a fixed phase function and its parameters, one way to implement the optimization is using the (μ_t , α) parameterization of voxels. However, a discretization step is then needed to turn the optimized volume into a printable specification. Because the printer gamut is smaller than what this voxel-parametrization can describe, it also implies the need for gamut mapping.

Other approaches incorporate the discretization step into the optimization loop (e.g., Elek et al. [2017]), guiding the optimizer through the limited gamut of printable medium parameters. Since our approach requires end-to-end derivatives, it would require calculating derivatives of the discretization step.

We propose an alternative solution that maintains differentiability and adjusts the volume properties in a space that is directly transformable into a material configuration. Assuming for the moment that the printer mixes the inks prior to placing them, we can express the optical properties of a voxel as a continuous affine combination of these materials. We call this the *material-space parameterization* of the medium:

$$V' = (\rho_{j,1}, \rho_{j,2}, ..., \rho_{j,N}) \quad \sum_{n=1}^{N} \rho_{j,n} = 1, \rho_{j,n} > 0, \qquad (7)$$

Table 2. Material Parameters. Materials in **bold** represent the Stratasys "Vero Opaque Rigid" Family, as measured by Elek et al. [2017]. The remaining lines show hypothetical materials used for evaluation. The extinction-coefficient μ_t and the single scattering albedo α are given in the three Red, Green and Blue color channels. All materials are using a Heney-Greenstein phase function with q = 0.4

	$\mu_t [mm^{-1}]$			α		
Material	R	G	В	R	G	В
Cyan	9.0	4.5	7.5	0.05	0.7	0.98
Cyan2	6.0	3.0	5.0	0.05	0.95	0.98
Magenta	2.5	3.0	10.0	0.98	0.1	0.9
Magenta2	1.7	2.0	6.7	0.98	0.05	0.97
Yellow	2.25	3.75	19.0	0.997	0.995	0.15
Yellow2	3.0	5.0	19.0	0.92	0.92	0.15
Orange	3.0	4.0	15.0	0.98	0.32	0.114
Black	5.0	5.5	6.5	0.35	0.35	0.35
White	6.0	9.0	24.0	0.9991	0.9997	0.999
Transparent	10^{-4}	10^{-4}	10^{-4}	1.0	1.0	1.0

where *N* is the number of available printing materials and $\rho_{j,n}$ their weights in the mixture for voxel *j*. The materials are parameterized by their absorption and scattering coefficients, and their phase function, so that $M_n = (\mu_{a,n}, \mu_{s,n}, \phi_n)$. An example instance of printing material parameters, as they were measured by Elek et al. [2017], is shown in Table 2. This idea of parametrizing volumetric light transport using pre-set materials is analogous to the material mixture method presented by Gkioulekas et al. [2013].

Gradients of the loss function with respect to this material space parametrization can be obtained by applying the chain rule:

$$\frac{\partial e}{\partial V'} = \frac{\partial e}{\partial V} \frac{\partial V}{\partial V'}.$$
(8)

Such a volume parametrization allows the optimizer to interpolate within the whole range of the materials' properties to achieve an appearance match. As we will show in Sec. 4.7, it also allows for a simple discretization to ensure manufacturability. Finally, it allows us to easily adapt to different printing materials, without any precalculation and with negligible overhead in computation time.

4.4 Relation of material space and optical parameters

3D printing uses a subtractive color mixing system similar to 2D printing. In addition to the cyan, magenta, yellow, and black (CMYK) inks, a white substrate (W) and potentially a transparent material (T) are available. Mathematically, an affine combination of these base materials has absorption and scattering characteristics that can be computed as a linear combination. Please note that this is only possible using the absorption- and scattering-coefficients, since their behaviour is linear. In contrast, a weighted sum of α and μ_t does not work the same way.

Let *z* describe this mapping from the (continuous) material space $(\rho_1, \ldots, \rho_N) \in \mathbb{M}$ to the space $(\mu_t, \alpha) \in \mathbb{S}$. Omitting the voxel index *j* for readability, we define it as:

$$z: \mathbb{M} \to \mathbb{S} = \left(\sum_{n=1}^{N} \rho_n(\mu_{s,n} + \mu_{a,n}), \frac{\sum_{n=1}^{N} \rho_n \mu_{s,n}}{\sum_{n=1}^{N} \rho_n(\mu_{s,n} + \mu_{a,n})}\right).$$
(9)

We assume all materials share the same phase function (Henyey-Greenstein, g = 0.4). Phase functions mix in the following way [Gkioulekas et al. 2013]:

$$\phi(\cos\theta) = \frac{\sum_{n=1}^{N} \rho_n \mu_{s,n} \phi_n(\cos\theta)}{\sum_{n=1}^{N} \rho_n \mu_{s,n}}.$$
 (10)

It can be easily seen that our assumption of equal phase functions for all materials results in the mixed phase function being the same as well.

Material space can be perceptively ambigous in several ways. First, and depending on the printing materials chosen, there can be many combinations in \mathbb{M} leading to the same optical properties in \mathbb{S} . As an example, consider mixing a single voxel of gray color from CMYKW. Even if CMY were perfectly orthogonal, it can be mixed in several ways: either by an appropriate ratio of black and white ink, or by replacing black with an even mixture of cyan, magenta and yellow, combining their absorption. The material densities can differ, so possible solutions for a desired color can have different *scattering* behavior, while having the same single-scattering *albedo*. This creates a mixing ambiguity that is more complex than in 2D printing.

Second, creating a specific appearance is *locally* ambigous even within space S. This was formalized with *similarity theory* [Wyman et al. 1989a,b; Zhao et al. 2014], which studies the equivalence classes of the volumetric parameters in radiative transfer, where different sets of parameters can lead to equivalent measurements of the radiance field. And finally, heterogeneous media can also have *nonlocal* ambiguities, following the observation that in such a medium, different spatial distributions of materials can lead to similar measurements [Gkioulekas et al. 2016].

Our parametrization helps the optimizer navigate these ambiguities. Following the example of mixing a gray color from above, the black ink will have a larger gradient value than each of the individual CMY inks, so the optimizer will favor the use of black ink for the respective voxel. This can be seen, for example, in Fig. 1,



Fig. 4. Comparison of the results optimized with different loss functions on this 4 cm tall 3D-scanned model. We can see that only relying on a ΔE_{76} -based loss function (Eq. 11) leads to a good overall color fidelity, but poor contrast in details, such as in the eye region or the necklace. That can be significantly improved with an introduction of 3D voxel-based loss function that take the structure into account as well. Similar behavior can be seen in our slab experiments in Figs. 1 and 19.

where the optimizer favors the use of a newly introduced orange ink for the creation of the orange patches and the red flowers in image Fig. 1(c). Further, assuming identical phase functions for all materials reduces the local ambiguities described by similarity theory. Empirical evidence [Gkioulekas et al. 2016, 2013; Zhao et al. 2016b] also suggests that this parametrization can be used in inverse rendering to recover ground-truth scattering properties in spite of the ambiguities.

Our proposed pipeline does not require computing the inverse of *z*, but only partial derivatives with respect to affine weights ρ . In practice, we implement the computation of gradients of the form $\partial V/\partial V'$ (Eq. 8, second factor) by auto-differentiating the material mixing function z(V'), Eq. 9. Together with $\partial e/\partial V = z'(V')$, one obtains an end-to-end derivative chain of the optimized function as visualized in Figure 3.

4.5 Error metrics

Our framework is driven by minimizing the loss function E(C, T). It governs the extent with which each visual stimulus affects the final results. Based on the exploration of existing image difference metrics in Sec. 2.3, we want to find a loss function for our use case with the following qualities:

- *differentiable*, in order to compute $\partial E/\partial C$,
- without discontinuities affecting the optimizations convergence,
- *perceptual*, to take the human visual system into account, considering *color fidelity, contrast*, and *structure*, and
- geometry-aware, to preserve local features.

In Fig. 4, we illustrate how significant the difference can be between the convergence with a simple error metric, compared to our more

sophisticated metric combination. The remainder of this subsection discusses how we built this single well-behaved metric by first selecting suitable 2D metrics and then generalizing them to voxelized 3D surfaces.

Color fidelity. In order to steer the optimization towards maximum color fidelity, we use the mean squared error in the CIELAB color space, inspired by ΔE_{76} (CIE dE 76). Note that MSE-style errors like ΔE_{76} are computed per-pixel (per-texel) and are invariant to the local neighborhood, hence they can be directly used in 3D:

$$E_{1}(C,T) = \frac{1}{I} \sum_{i=1}^{I} \left(\left(c_{i,L^{*}} - t_{i,L^{*}} \right)^{2} + \left(c_{i,a^{*}} - t_{i,a^{*}} \right)^{2} + \left(c_{i,b^{*}} - t_{i,b^{*}} \right)^{2} \right), \quad (11)$$

where indices L^* , a^* , and b^* denote $L^*a^*b^*$ components of the respective texel.

Contrast and structure. E_1 is a global color metric that is not sensitive to visual stimuli such as local contrast and structure. To improve on that, we employ a normalized structural similarity index measure (SSIM), computed over local neighborhoods with a Gaussian sliding window and convolutions [Orihuela and Ebrahimi 2019].

While SSIM was originally designed for 2D signals, we generalize it to 3D as follows: Let P_{c_i} and P_{t_i} denote local neighborhoods of the *i*-th predicted and target texels. Let G_{σ} denote a 3D Gaussian kernel with a standard deviation σ . The mean intensities $\bar{c_i}$ and $\bar{t_i}$, standard deviations σ_{c_i} and σ_{t_i} , and covariance $\sigma_{c_i t_i}$ are then given by:

$$\bar{c}_i = G_\sigma * P_{c_i}, \qquad \sigma_{c_i}^2 = G_\sigma * P_{c_i}^2 - \bar{c_i}^2,$$
(12)

$$\bar{t}_i = G_\sigma * P_{t_i}, \qquad \sigma_{t_i}^2 = G_\sigma * P_{t_i}^2 - \bar{t_i}^2,$$
 (13)

$$\sigma_{c_i t_i} = G_{\sigma} * \left(P_{c_i} \cdot P_{t_i} \right) - \bar{c_i} \bar{c_t}, \qquad (14)$$

where * denotes the discrete convolution operator. For the *i*-th texel neighborhood, local SSIM is then computed as:

$$SSIM(c_i, t_i) = \frac{2\bar{c_i}\bar{t_i} + K_1}{\bar{c_i}^2 + \bar{t_i}^2 + K_1} \cdot \frac{2\sigma_{c_it_i} + K_2}{\sigma_{c_i}^2 + \sigma_{t_i}^2 + K_2},$$
(15)

where $K_1 = 0.01^2$ and $K_2 = 0.03^2$ are constants ensuring numerical stability. The final loss function $E_2(C, T)$ is then computed as a mean over all local neighborhoods, considering the three L*a*b* components separately:

$$E_2(C,T) = 1 - \frac{1}{I} \sum_{i=1}^{I} \sum_{L^* a^* b^*} SSIM(c_i, t_i).$$
(16)

For computing E_2 , we experimentally observed better behaviors with the L*a*b* components being normalized to [0, 1].

Edge similarity. As Chen et al. [2006] observed, SSIM prefers blurred data over noisy data. This poses a problem, since we use a Monte Carlo estimator with inherent variance. They propose to modify SSIM with the Sobel operator used for edge detection, denoting their metric as ESSIM.

Here, we use a simpler error E_3 based on MSE between the texels with a Sobel operator applied. S_x , S_y , and S_z denote the 3D Sobel kernels along the x, y, and z axes, respectively. The convolution between the kernels and *C* or *T* approximate the derivatives of the texels' signal along the three axes. Computing an MSE over the results in each of the three axes separately, and doing the same with the $L^*a^*b^*$ components, gives our third metric:

$$E_3(C,T) = 100 \cdot \frac{1}{3} \sum_{x,y,z} \sum_{L^* a^* b^*} \frac{1}{I} \frac{1}{4} \frac{1}{32} \sum_{i=1}^{I} (S * C - S * T)^2.$$
(17)

The normalization constants are given by the number of axes (3), the minimum and maximum values in the convolution (-32 and 32), and the maximum value after the square $((-1 - 1)^2 = 4)$. The additional factor of 100 is used to ensure the typical error values are in the similar range as those of E_1 and E_2 , which leads to a more intuitive weighting in the final metric. Note that similarly to E_2 , we are considering the normalized L*a*b* color space.

Final metric. We obtain the final loss function E(C, T) as a weighted sum of the proposed metrics. To ensure that changing the individual weights does not change the overall magnitude of the error value, we define the sum such that the weights are always normalized:

$$E(C,T) = \frac{\gamma_1}{\gamma} E_1(C,T) + \frac{\gamma_2}{\gamma} E_2(C,T) + \frac{\gamma_3}{\gamma} E_3(C,T),$$
 (18)

where γ_1 , γ_2 , and γ_3 are the weights of E_1 , E_2 , and E_3 , respectively, and $\gamma = \gamma_1 + \gamma_2 + \gamma_3$. In Fig. 19, we illustrate how changing the individual weights influences the result. Experimentally, we observed that the highest overall color accurancy and saturation are achieved by setting $\gamma_2 = \gamma_3 = 0$. However, that leads to a loss of contrast and details, which may be significant in certain situations, such as the Nefertiti model in Fig. 4. Hence, we recommend using $\gamma_1 = 1$, $\gamma_2 \in [3, 9]$, and $\gamma_3 \leq 3$, depending on the desired appearance. Higher values may lead to an unpleasant loss of saturation, in case of γ_2 , or substantial color casts, in case of γ_3 .

4.6 Optimization algorithm

We implemented the optimization using the TensorFlow framework [Abadi et al. 2015]. The renderer is implemented as a function into the framework, using the custom gradient decorator. Ada-Grad [Duchi et al. 2011] is used as the optimization algorithm, but also experimented with a simple gradient descent with momentum and did not notice any significant differences. We constrain the optimizer to positive values using the sigmoid function. A successive per-voxel normalization of the ink mixture vector constrains the result to the feasible set.

This is in contrast to projected gradient descent (e.g., [Duchi et al. 2008]), where Euclidean projection onto the feasible set is used after applying the gradient in each step. It is also possible to use exponentiated gradient descent (Shem-Tov et al. [2020] show a domain-specific application). Both algorithms perform favorably if only few components of the inputs are relevant for the loss, which is the case for bigger volumes. There, voxels far beneath the surface have negligible influence on surface appearance. We see this as an interesting direction for future work.

4.7 Probabilistic ink quantization

The material space \mathbb{M} parametrization introduced above assumes a printhead that physically mixes inks within the printed volume.

178:8 • Nindel et al

In reality, however, the printer can only deposit one material per addressable voxel and relies on spatial mixing together with lateral light transport to blend colors. Thus, after the optimization loop, the resulting volumes V' with affine weights require discretization to obtain a valid printer input.

Existing halftoning or contoning approaches [Babaei et al. 2017; Brunton et al. 2015; Sumin et al. 2019] could be applied directly, but without their color mapping as we operate directly in material space. We propose a simple solution based on random sampling, exploiting the fact that printers can reach significantly higher resolutions than the 300 DPI we run our optimization in. While upsampling to the printer resolution, we stochastically assign a single material to each voxel based on its affine weights: For each (micro-)voxel in the higher-resolution quantized grid, an ink is drawn at random, using the corresponding (macro-)voxel's material mixture as the probability distribution. At a native resolution of $600 \times 300 \times 940$ DPI, this gives a ratio of about 1 : 6 between the number of voxels in the optimized and discretized grids of the printer. Figure 5 visualizes this procedure in a 2D example where a lower-resolution grid is upsampled to yield a discrete printer assignment.

A downside of this simple solution is that random sampling is known to introduce white noise. To some extent, this is mitigated by the fact that the scattering medium acts as a low pass filter that removes high frequency components, the cutoff frequency being a function of the opaqueness of the materials. However, some of this noise is still visible on the optimized objects, seen in Figs. 10 and 11 as graininess on the object's surface. In Fig. 6 we show a side-byside comparison of a continuous material space and corresponding probabilistically quantized versions: we see that the remaining noise has a small impact on the results. In the future, implementing more sophisticated halftoning techniques, like the ones mentioned above, can help eliminate the noise introduced by random sampling.

4.8 Simulations vs. real printouts

Our printer model assumes perfectly disjunct, cubical voxels that are each filled with exactly one material. In contrast, the actual printing process is subject to a variety of imperfections. Physical mixing of adjacent voxels can occour around their boundaries.

This is especially noticable on the printout's surface, where support material and colored inks interact. To counter this, prints can be done with thin wrap of transparent material, that is subsequently removed by abrasion. This will also remove remnants of support



Fig. 5. A diagram of a discrete grid with continuous ink mixtures (left) and its corresponding probabilistically quantized ink arrangement (right). In this example, the quantized grid has two times higher resolution in the horizontal axis, and three times higher in the vertical axis, which roughly corresponds to the XZ axes of our $600 \times 300 \times 940$ DPI quantized grid.

ACM Trans. Graph., Vol. 40, No. 4, Article 178. Publication date: August 2021.



Fig. 6. Comparing renderings of a solution with continuous material mixtures (left, 300 DPI) to probabilistically quantized results (right, $600 \times 300 \times$ 940 DPI) for a 1 mm planar target. Despite different random seeds, about 70% of the quantized voxels are *identical*; the variance does not cause any substantial visual difference.

material on the surface. A coat of transparent paint establishes the smooth dielectric surface we assume in the optimization.

Inside the prinout, inks from adjacent voxels can flow into each other before UV light is applied to harden them. The resulting mixture of pigments creates a new material, which can be modelled under the assumptions given in Sect. 4.4. Since the physical mixing occours at a higher spatial frequency than we model the volume at, the simulations effectively under-sample the physical printout. Fig. 6 helps quantify the impact of this effect: On the left is a rendering of a continous mixture mapped into absorption/scattering space using Eq. 9 at 300 DPI, simulating perfect physical mixing inside each voxel. The images on the right side of the figure show renderings with voxels of discrete materials, at a higher spatial resolution. No aliasing artifacts are visible.

Assuming an identical phase function for all material also introduces differences between simulation and real printouts, because the phase functions of the actual materials are not identical. Different but optically equivalent mixtures in simulation will have small differences in their appearance when printed. Lastly, using RGB (as opposed to spectral rendering) to handle color throughout the pipeline limits the accuracy of the simulation of the participating medium, introducing color reproduction artifacts. Some effects in participating media can be simulated correctly only by a spectral workflow. One such effect is spectral sharpening, where the repeated application of an absorption spectrum results in a narrowing of its peaks.

Inverse rendered solutions depend on the assumed lighting conditions. This means that under different lighting than those used during optimization, the resulting medium may actually have incorrect appearance. If the display conditions of the printout are known beforehand, this lighting can be "baked" into the optimized result by replicating it in the renderer. Otherwise, assuming hemispherical illumination provides a good default-case.

5 EVALUATION

In the following, we show results obtained using the presented pipeline and compare them to other solutions.

Setup. As both Elek et al. [2017] and Sumin et al. [2019] demonstrated the faithfulness of predictive renderings in comparison to

physical printouts, we rely on virtual simulations to evaluate our method. We see this as a viable way, as all our figures depict quantized solutions that could be directly fabricated. For the optical properties of inks, use the measurements by Elek et al. [2017] for the Vero Rigid Opaque materials used in the Stratasys J7/J8 3D printer family. Alongside, we designed hypothetical materials (Table 2) to showcase our pipeline's flexibility. For discretization we use a distance-field based voxelization process with uniform 300 DPI resolution as presented in [Sumin et al. 2019]. Our renderer assumes a uniform surface illumination with unity radiance. The sensor is in texel-space, i.e., each fragment of the sensor corresponds to exactly one voxel on the surface. Radiative backpropagation is used with the $L_i = 1$ approximation. We initialize the optimization process with a 97% white mixture for all voxels, the remainder is split equally into the other inks to make sure their initial gradients are non-zero. For the final quantization, the resolution is increased to a typical printer resolution of $600 \times 300 \times 940$ DPI. We model the surface of all objects as a smooth dielectric with an refractive index of 1.5.

5.1 Opaque Volume Appearance

We apply our framework to recover the opaque, textured appearance of a scattering 3D print and demonstrate our results in comparison to previous work along this line of research.

Contrast and spatial resolution. Starting with a basic property, we design a test pattern to evaluate edge contrast and lateral scattering compensation. The pattern features black and white stripes of varying widths, with the smallest being one voxel (about 85 µm). As shown in Fig. 7, a naïve placement of black and white ink results in a well-preserved lightness match on larger patches at the cost of colorbleeding around edges. Our solution stays color-neutral with slightly increased lightness deviation in flat regions. Despite variance from multiple sources, our optimization reproduces fine details accurately and balances local contrast, spatial details, and color-bleeding. The cutouts in Fig. 8 enlarge a single step-function from Fig. 7. Upon closer inspection, one can notice that our optimizer decides to mix a substantial amount of colored ink into the black regions, mainly



Fig. 8. Rendering of a discretized optimization result for a step function; (a) artificially increased density to show individual voxels, (b) actual appearance.

magenta, and yellow ink into the white region. This counters the slight tints of the white and black materials and ensures the result stays color neutral, unlike the naïvely extruded solution. Such shortcuts are possible with color-table-based approaches [Babaei et al. 2017; Brunton et al. 2015; Elek et al. 2017], but require manual discovery, an explicit specification, and an extension of the mapping. Our method discovers and selectively applies these alternative solutions without intervention, while explicit choice would have to be incorporated in the metric.

Color and structure fidelity. Testing color reproduction, we run the optimization on textured slabs which show a combination of smooth color gradients, hard edges, and fine details. Fig. 9 shows the performance of the proposed method in comparison with the state of the art [Sumin et al. 2019].

In the top image, we observe overall higher saturation for our method. Fine details around the table, chair and painting are slightly more pronounced on the left side. The structure of the floor is however more refined for the previous method.

Looking at the bottom image, they perform better at reproducing the grainyness in the sky. When comparing the challenging case of black-on-white details on the cottage's wall, which have the extend of single voxels, the methods seem to be almost on par. We note, that the dark-orange spots along the path are difficult to reproduce, as also visible in the cutouts in Fig. 1.

Metric weighting. In Sec. 4.5, we proposed a new combined metric for 3D applications that balances color accuracy, local structure, and



Naïve KW extrusion

Fig. 7. Testing maximum resolution and color fidelity of a 1 mm thick planar target with a "zebra" test pattern (left). The pattern contains black and white stripes of increasing frequency (stripe heights: 30, 15, 10, 5, 3, and 1 voxels; 1 voxel is about 85 µm at 300 DPI). Our method (top) reproduces even the tiniest details without any moiré fringes or significant color shifts. A naïve solution (bottom) consisting of black (K) and white (W) inks extruded through the whole slab results in slightly better contrast, but noticeable color shift towards blue and over-darkening of the right part of the slab. To verify, difference images in the CIELAB color space are attached (right), computed as Euclidean distances in L*, a*, and b* separately.



Fig. 9. Comparison of our results (left) to the state of the art (right) [Sumin et al. 2019] for a 2.5D planar slab, which has a different texture on each side. The slab is 30x20 mm large and 2.5 mm thick. **Top:** On the first side of the slab, our method achieves a significantly better color accuracy, saturation, and contrast than the state of the art. **Bottom:** The differences on this side are more subtle: our result overall appears "smoother" and managed to reproduce the dark edges better.

edge sharpness. Using different weighting for the components E_1 , E_2 , and E_3 , one can steer the optimization trajectory. The parameterspace exploration shown in Fig. 19 adjusts the relative weights for the latter two, while keeping E1 (color accuracy) constant.

We observe that increasing values of E_2 in the first row leads to higher contrast around edges, as can be seen in the outlines of the chair and bed on the left side. This also leads to desaturation of colors for example on the blue wall.

Increasing the weight for E_3 leads to even stronger contrast, but at the cost of severe hue shifts towards the higher values (bottom row). The combination of all three metrics towards the bottom right inherits the properties of individual components.

Co-optimizing different visual stimuli seems to lead to them influencing one another. This is likely because maximum contrast and best color fidelity are conflicting goals: A black to white step function has maximum contrast, but no color.

ACM Trans. Graph., Vol. 40, No. 4, Article 178. Publication date: August 2021.

Table 3. Summary of optimization results. We use three sets of hardware: \Box denotes a NVIDIA RTX 3080 (10 GB) and * means using six materials instead of five. \natural runs on a NVIDIA TITAN RTX (24 GB). [Sumin et al. 2019] is run on a machine with 20 cores (non-HT) \dagger .

Model	Size	Volume voxels	Surface voxels	Total runtime	Samples	Setup
Orange	35 mm	3.70 M	359 k	215 min	128 spp	•*
Slab	15 mm	0.69 M	118 k	50 min	128 spp	o
	30 mm	2.73 M	405 k	424 min	128 spp	·
				430 min	128 spp	•*
				342 min	512 spp	†
Cat	60 mm	8.62 M	637 k	1350 min	128 spp	4
				1429 min	512 spp	†
Nefertiti	40 mm	10.50 M	576 k	1042 min	32 spp	4
				3071 min	512 spp	†

General 3D geometry. Combining our previous results into a practical setting, we run our framework on several objects with general 3D geometry. Fig. 1 (right) shows them side-by-side to illustrate their scale. The cat model shown in Fig. 11 was selected for its complex curvature, thin features such as the ears and tail, and a highly detailed face texture. A 3D scan of Nefertiti was chosen for its very thin, but high-contrast details such as the eyebrows, eyeliner, and necklace area. Optimization results and a comparison to the state of the art is shown in Fig. 10.

Computational cost. All models shown were run on consumergrade hardware equipped with either the NVIDIA TITAN RTX (24 GB memory) or 3080 RTX (10 GB memory) GPU. Table 3 details runtimes and setups used for individual models and shows that the total runtimes of our pipeline and the method of Sumin et al. [2019] seem roughly comparable, though a GPU-based implementation of their method would likely shift this intuition in their favor. That already takes into account that we adjusted our learning rate to achieve good convergence within about 150-350 iterations, whereas the heuristic pipeline of Sumin et al. [2019] usually reaches the lowest error in 10 steps.

In Fig. 18, we illustrate that our runtimes scale roughly linearly with respect to model sizes. In Fig. 17, we show how our method can be adjusted in a "time vs. quality" manner, where decreasing the sample counts results in faster runtimes, but lower quality of details.

Initialization. It can be suspected that the optimized objective function contains many local minima, most of them having very similar values. With gradient descent being a local optimization method, convergence to the absolute global minimum cannot be guaranteed, but we can show that even different initializations still converge to an almost identical error value. Results of this initialization study can be found in Figs. 12, 13 and 14 for the 30 mm × 20 mm slab model. Initializing with results obtained from Sumin et al. [2019]



Fig. 10. A 4 cm tall reproduction of a 3D-scanned model of Nefertiti. Difficult areas are the eye region, where our method shows better detail, and the necklace, where Sumin et al. [2019] excels in contrast



Fig. 11. Cat model: Its orange-brown texture seems to be especially tough to match. Notice that our result is slightly darker, towards brown, while Sumin et al. [2019] are slightly lighter.



Fig. 12. Convergence plots of different initializations using the $30\,\mathrm{mm}\times20\,\mathrm{mm}$ slab model.



Fig. 13. Comparing different initializations of the 30 mm slab model. (a) our default initialization with a 97% white mixture, (b) initialized with the converged result from [Sumin et al. 2019]

greatly accelerates the convergence of the optimization. Without this informed start, the fastest convergence is obtained when initializing with white, while a random mixture needs more iterations to converge. Inspecting the results reveals small differences. Initializing the mixture with random values seems to result in mixtures with less black, suggesting that the optimizer chooses to favor darkening by using a combination of CMY. The resulting images show higher saturation, but lower contrast. Initializing the pipeline with a solution from Sumin et al. [2019] shows no significant differences to the white initialization. We investigated this further by studying differences in L,a,b seperately, and found the biggest delta in the sky lightness in the scene of Fig. 13.

5.2 Applications

A key feature of the proposed pipeline is the ease of tailoring it to different use cases. By changing the behavior of its individual components, the objective of the optimization can be adapted. In the following, we show how two previously complex tasks that required specialized solutions naturally follow from our parametrization.

Extended Ink-set. Introducing a new (specialized) printing material is useful for increased color accuracy and spectral reproduction [Shi et al. 2018], controlling reflectance [Piovarči et al. 2020], and co-optimizing mechanical and optical parameters [Morovič et al. 2019]. In our pipeline, the optical properties of new materials are



Fig. 14. Average ink usage of the top 8 layers of the lower half of the 30mm slab model (Fig. 13), different initializations



Fig. 15. Material-selection use-case for a printer that can use 6 different materials. Running all possible $\binom{6}{4}$ combinations of the first 6 inks from Tab. 2, plus Black and White, reveals the combination Cyan, Magenta, Magenta2, Yellow having the lowest loss value and thus being the inkset best suited for reproducing the target on the printer.

simply added to the available set with material mixing (Eq. 9) and quantization (Sec. 4.7) adjusted accordingly. An example of this can be seen in Fig. 1, where an additional sixth, orange material is being used to improve the color on the pathway. The associated time penalty is negligible as shown in Table 3. This is due to the fact that rendering and computing the light-transport derivatives always takes place in RGB. The material mixture is handled before the rendering, and after radiative backpropagation, so the associated memory bandwidth impact is very small. Previous methods required expensive rebuilding or measuring of color tables whereas our method adapts to new material combinations on the fly.

Furthermore, because of this flexibility, it becomes feasible to tackle the long-standing *ink selection* problem from 2D printing. This is analogous to hyperparameter-optimization in machine learning where an outer loop is optimizing the meta-parameters of the inner training loop.

The convergence graph in Fig. 15 reveals a certain optimal ink-set for the 15 mm slab model shown in Fig. 1. For this proof-of-concept we show a brute-force search but note that more sophisticated strategies (e.g., Shi et al. [2018] and Ansari et al. [2020]) could be transferred from the aforementioned fields in the future.

Translucency. A special case of an extended material set is the addition of transparent ink. Brunton et al. [2018] perform informed, probabilistic replacement of white ink with transparent material to reproduce perceptual transparency cues. In a similar way, we can constrain the optimizer to a use a fixed amount of transparent material in order to enforce translucency in the optimization result.



Fig. 16. Translucency reproduction of an orange slice. (a) shows the optimization results using the opaque material set and a dE76 loss function. In (b) the optimizer is constrained to 80% transparent ink for the pulp.

One such example can be seen in Fig. 16, where the optimizer is constrained to 80% transparent material for the pulp of the orange, and was allowed to choose freely in the peel. This results in a more realistic appearance reproduction compared to the opaque version in Fig. 16a.

Translucent appearance is a perceptively highly complex phenomenon. The human visual system depends on certain visual cues to disambiguate the appearance of different materials, such as color saturation, edge sharpness and the appearance under side- or backlighting [Fleming and Bülthoff 2005; Xiao et al. 2014, 2020].

Future work can build on this and integrate spatially-varying constraints, or perceptive metrics that include translucency, to replicate artist-friendly control akin to Brunton et al. [2018] or directionallydependent goals as discussed by Zheng et al. [2020].

6 CONCLUSION

Resin-based full-color 3D-printing is capable of creating objects with very complex light transport properties. Voxels deep inside of the volume can have significant influence on the surface appearance of printouts, depending on how the surrounding volume is shaped and configured. Modeling these interdependencies on a global scale can give a degree of control over the results that was previously not attainable. Combining Monte Carlo estimates of light transport gradients from a physically based renderer with a volume parameterization based on the actual printer materials gives such a model. Its application in the context of gradient-descent optimization with a suitable loss function leads to results that come close to what is possible given the physical limits of the printing process.

The flexibility of this approach is illustrated by the fact that problems that previously required specialized solutions turn into straightforward extensions of the same pipeline. The extended inkset, inkset optimization and translucency use-cases have shown that. Our metrics parameters and setting the sampling rate for the MC estimates give intuitive controls over the optimization process on a "color vs. contrast" and "time vs. quality" basis.

The ambiguities that stem from the available materials can actively be exploited by the optimizer to attain a good match, instead of complicating the creation of pre-computed mapping tables. It

Further work still remains. The way the human visual system perceives different features proves to be very difficult to distill into a conclusive metric, as the field of image quality- and image-differencemetrics shows. Implementing new discoveries in this field into the pipeline will be easy, as long as they are differentiable.

Furthermore, with the optimization being gradient-based, it only explores a limited subset of the whole solution space. One can suspect that better solutions exist, such as building transparent channels inside the volume that would optically connect areas that are geometrically far from each other. But such complex configurations cannot be found by the optimizer without a good initialization.

ACKNOWLEDGMENTS

We thank Sketchfab users Evgeni Yanev (Orange Slice), C. Yamahata and Thot_htp (Nefertiti) for their 3D models, and the reviewers for their valuable feedback. This work has received funding from GA UK project 1164620 of Charles University, and by the Czech Science Foundation under grant number GAČR-19-07626S. This work was further supported by the Charles University grant SVV-260588.

REFERENCES

- Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. http://tensorflow.org/ Software available from tensorflow.org.
- Navid Ansari, Omid Alizadeh-Mousavi, Hans-Peter Seidel, and Vahid Babaei. 2020. Mixed integer ink selection for spectral reproduction. ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 39, 6 (Nov. 2020), 255:1–255:16. https://doi.org/10.1145/ 3414685.3417761
- Thomas Auzinger, Wolfgang Heidrich, and Bernd Bickel. 2018. Computational design of nanostructural color for additive manufacturing. ACM Transactions on Graphics (Proc. SIGGRAPH) 37, 4 (July 2018), 159:1–159:16. https://doi.org/10.1145/3197517. 3201376
- Vahid Babaei, Kiril Vidimče, Michael Foshey, Alexandre Kaspar, Piotr Didyk, and Wojciech Matusik. 2017. Color Contoning for 3D Printing. ACM Transactions on Graphics (Proc. SIGGRAPH) 36, 4 (July 2017), 124:1–124:15. https://doi.org/10.1145/ 3072959.3073605
- Atılım Günes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. 2017. Automatic Differentiation in Machine Learning: a Survey. Journal of Machine Learning Research 18, 1 (2017), 5595–5637.
- Alan Brunton, Can Ates Arikan, Tejas Madan Tanksale, and Philipp Urban. 2018. 3D Printing Spatially Varying Color and Translucency. ACM Transactions on Graphics (Proc. SIGGRAPH) 37, 4 (July 2018), 157:1–157:13. https://doi.org/10.1145/3197517. 3201349
- Alan Brunton, Can Ates Arikan, and Philipp Urban. 2015. Pushing the Limits of 3D Color Printing: Error Diffusion with Translucent Materials. ACM Transactions on Graphics 35, 1 (Dec. 2015), 4:1–4:13. https://doi.org/10.1145/2832905
- Chengqian Che, Fujun Luan, Shuang Zhao, Kavita Bala, and Ioannis Gkioulekas. 2020. Towards Learning-based Inverse Subsurface Scattering. In 2020 IEEE International Conference on Computational Photography, ICCP 2020, Saint Louis, MO, USA, April 24-26, 2020. IEEE, New York, NY, USA, 1–12. https://doi.org/10.1109/ICCP48838. 2020.9105209
- Guan-Hao Chen, Chun-Ling Yang, Lai-Man Po, and Sheng-Li Xie. 2006. Edge-Based Structural Similarity for Image Quality Assessment. In *IEEE International Conference* on Acoustics Speech and Signal Processing Proceedings, Vol. 2. IEEE, New York, NY, USA, 933–936.

- Yue Dong, Jiaping Wang, Fabio Pellacini, Xin Tong, and Baining Guo. 2010. Fabricating spatially-varying subsurface scattering. ACM Transactions on Graphics (Proc. SIG-GRAPH) 29, 4 (July 2010), 62:1–62:10. https://doi.org/10.1145/1778765.1778799
- John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research* 12, Jul (2011), 2121–2159.
- John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. 2008. Efficient Projections onto the L1-Ball for Learning in High Dimensions. In 25th International Conference on Machine Learning (Helsinki, Finland) (ICML '08). Association for Computing Machinery, New York, NY, USA, 272–279. https://doi.org/10.1145/ 1390156.1390191
- Oskar Elek, Denis Sumin, Ran Zhang, Tim Weyrich, Karol Myszkowski, Bernd Bickel, Alexander Wilkie, and Jaroslav Křivánek. 2017. Scattering-aware Texture Reproduction for 3D Printing. ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia) 36, 6 (Nov. 2017), 241:1–241:15. https://doi.org/10.1145/3130800.3130890
- Roland W Fleming and Heinrich H Bülthoff. 2005. Low-level image cues in the perception of translucent materials. ACM Transactions on Applied Perception (TAP) 2, 3 (2005), 346–382.
- J. R. Frisvad, S. A. Jensen, J. S. Madsen, A. Correia, L. Yang, S. K. S. Gregersen, Y. Meuret, and P.-E. Hansen. 2020. Survey of Models for Acquiring the Optical Properties of Translucent Materials. *Computer Graphics Forum* 39, 2 (2020), 729–755. https://doi.org/10.1111/cgf.14023 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14023
- Ioannis Gkioulekas, Anat Levin, and Todd Zickler. 2016. An Evaluation of Computational Imaging Techniques for Heterogeneous Inverse Scattering. In European Conference on Computer Vision. Springer, Berlin, Germany, 685–701. https: //doi.org/10.1007/978-3-319-46487-9
- Ioannis Gkioulekas, Shuang Zhao, Kavita Bala, Todd Zickler, and Anat Levin. 2013. Inverse volume rendering with material dictionaries. ACM Transactions on Graphics (TOG) 32, 6 (2013), 162:1–162:13. https://doi.org/10.1145/2508363.2508377
- Miloš Hašan, Martin Fuchs, Wojciech Matusik, Hanspeter Pfister, and Szymon Rusinkiewicz. 2010. Physical Reproduction of Materials with Specified Subsurface Scattering. ACM Trans. Graph. 29, 4, Article 61 (July 2010), 10 pages. https: //doi.org/10.1145/1778765.1778798
- Roman Hochuli, Samuel Powell, Simon Arridge, and Ben Cox. 2016. Quantitative photoacoustic tomography using forward and adjoint Monte Carlo models of radiance. *Journal of biomedical optics* 21, 12 (2016), 126004.
- Pramook Khungurn, Daniel Schroeder, Shuang Zhao, Kavita Bala, and Steve Marschner. 2015. Matching Real Fabrics with Micro-Appearance Models. ACM Trans. Graph. 35, 1 (2015), 1–1.
- Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. 2018. Differentiable monte carlo ray tracing through edge sampling. ACM Transactions on Graphics (TOG) 37, 6 (July 2018), 125:1–125:12. https://doi.org/10.1145/3306346.3322954
- M Ronnier Luo, Guihua Cui, and Bryan Rigg. 2001. The development of the CIE 2000 colour-difference formula: CIEDE2000. Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur 26, 5 (2001), 340–350.
- A. Luongo, V. Falster, M. B. Doest, M. M. Ribo, E. R. Eiriksson, D. B. Pedersen, and J. R. Frisvad. 2020. Microstructure Control in 3D Printing with Digital Light Processing. *Computer Graphics Forum* 39, 1 (2020), 347–359. https://doi.org/10.1111/cgf.13807 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13807
- M. Magnor, G. Kindlmann, N. Duric, and C. Hansen. 2004. Constrained inverse volume rendering for planetary nebulae. In *IEEE Visualization 2004*. IEEE, New York, NY, USA, 83–90. https://doi.org/10.1109/VISUAL.2004.18
- K McLaren. 1976. XIII The development of the CIE 1976 (L* a* b*) uniform colour space and colour-difference formula. *Journal of the Society of Dyers and Colourists* 92, 9 (1976), 338–341.
- Peter Morovič, Ján Morovič, Ingeborg Tastl, Melanie Gottwals, and Gary Dispoto. 2019. Co-optimization of color and mechanical properties by volumetric voxel control. *Struct Multidisc Optim* 60, 3 (Sept. 2019), 895–908. https://doi.org/10.1007/s00158-019-02240-8
- Merlin Nimier-David, Sébastien Speierer, Benoît Ruiz, and Wenzel Jakob. 2020. Radiative Backpropagation: An Adjoint Method for Lightning-Fast Differentiable Rendering. *Transactions on Graphics (Proceedings of SIGGRAPH)* 39, 4 (July 2020), 146:1–146:15. https://doi.org/10.1145/3386569.3392406
- Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. 2019. Mitsuba 2: A Retargetable Forward and Inverse Renderer. *Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 38, 6 (Dec. 2019), 203:1–203:17. https://doi.org/10.1145/3355089. 3356498
- Isabel Molina Orihuela and Mehran Ebrahimi. 2019. An Efficient Algorithm for Computing the Derivative of Mean Structural Similarity Index Measure. In *Image Analysis* and Recognition, Fakhri Karray, Aurélio Campilho, and Alfred Yu (Eds.). Springer International Publishing, Cham, 55–66.

178:14 • Nindel et al



Fig. 17. Illustrating how the convergence can be adjusted in a "time vs. quality" manner. A $30 \text{ mm} \times 20 \text{ mm} \times 2.5 \text{ mm}$ double-sided slab, identical to the one in Fig. 9, was optimized with the same learning rate, but different sample counts per texel (spp). The ΔE_{76} -based metric was used for simplicity, so $\gamma_2 = \gamma_3 = 0$. Notice that lowering spp results in proportionally faster iterations, but lose details due to the noise in the Monte Carlo estimates of the derivatives. Despite the 32 spp optimization being almost four times faster, the differences will be almost negligible when looking at the object from further away.



Fig. 18. Time per iteration vs. volume for a spherical model (128 spp, 3080RTX)

- Marios Papas, Christian Regg, Wojciech Jarosz, Bernd Bickel, Philip Jackson, Wojciech Matusik, Steve Marschner, and Markus Gross. 2013. Fabricating translucent materials using continuous pigment mixtures. ACM Transactions on Graphics (TOG) 32, 4 (2013), 1–12.
- Michal Piovarči, Michael Foshey, Vahid Babaei, Szymon Rusinkiewicz, Wojciech Matusik, and Piotr Didyk. 2020. Towards spatially varying gloss reproduction for 3D printing. ACM Transactions on Graphics (TOG) 39, 6 (2020), 1–13.
- Jens Preiss, Felipe Fernandes, and Philipp Urban. 2014. Color-image quality assessment: From prediction to optimization. *IEEE Transactions on Image Processing* 23, 3 (2014), 1366–1378.
- ACM Trans. Graph., Vol. 40, No. 4, Article 178. Publication date: August 2021.

- Olivier Rouiller, Bernd Bickel, Jan Kautz, Wojciech Matusik, and Marc Alexa. 2013. 3D-printing spatially varying BRDFs. *IEEE computer graphics and applications* 33, 6 (2013), 48–57.
- Gaurav Sharma, Wencheng Wu, and Edul N Dalal. 2005. The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur 30, 1 (2005), 21–30.
- Kfir Shem-Tov, Sai Praveen Bangaru, Anat Levin, and Ioannis Gkioulekas. 2020. Towards Reflectometry from Interreflections. In 2020 IEEE International Conference on Computational Photography (ICCP). IEEE, New York, NY, USA, 1–12. https: //doi.org/10.1109/ICCP48838.2020.9105251
- Liang Shi, Vahid Babaei, Changil Kim, Michael Foshey, Yuanming Hu, Pitchaya Sitthi-Amorn, Szymon Rusinkiewicz, and Wojciech Matusik. 2018. Deep multispectral painting reproduction via multi-layer, custom-ink printing. ACM Trans. Graph. 37, 6 (Dec. 2018), 1–15. https://doi.org/10.1145/3272127.3275057
- Denis Sumin, Tobias Rittig, Vahid Babaei, Thomas Nindel, Alexander Wilkie, Piotr Didyk, Bernd Bickel, Jaroslav Křivánek, Karol Myszkowski, and Tim Weyrich. 2019. Geometry-Aware Scattering Compensation for 3D Printing. ACM Transactions on Graphics (Proc. SIGGRAPH) 38, 4 (July 2019), 111:1–111:14. https://doi.org/10.1145/ 3306346.3322992
- Philipp Urban, Tejas Madan Tanksale, Alan Brunton, Bui Minh Vu, and Shigeki Nakauchi. 2019. Redefining A in RGBA: Towards a Standard for Graphical 3D Printing. ACM Trans. Graph. 38, 3 (June 2019), 21:1–21:14. https://doi.org/10.1145/3319910
- Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image* processing 13, 4 (2004), 600–612.
- Zhou Wang, Eero P Simoncelli, and Alan C Bovik. 2003. Multiscale structural similarity for image quality assessment. In The Thrity-Seventh Asilomar Conference on Signals,

Relative weight of *E*₂ (contrast and structure) Relative weight of *E*₂ (contrast and structure) (http://opencession.opence

Fig. 19. Illustrating how the weights γ_2 and γ_3 of our error metrics E_2 and E_3 influence optimization results on a miniature 15x10x2.5 mm double-sided slab. Note that we fixed $\gamma_1 = 1$. All results are quantized, the result of Sumin et al. [2019] is shown for reference. Notice how increasing the SSIM weight γ_2 results in a higher contrast and structure, but also more "washed-out" colors. The strongest edges can be achieved with a high γ_3 , at the cost of a very poor color fidelity.

Systems & Computers, 2003, Vol. 2. IEEE, New York, NY, USA, 1398-1402.

- Douglas R Wyman, Michael S Patterson, and Brian C Wilson. 1989a. Similarity relations for anisotropic scattering in Monte Carlo simulations of deeply penetrating neutral particles. J. Comput. Phys. 81, 1 (1989), 137–150.
- Douglas R Wyman, Michael S Patterson, and Brian C Wilson. 1989b. Similarity relations for the interaction parameters in radiation transport. *Applied optics* 28, 24 (1989), 5243–5249.
- Bei Xiao, Bruce Walter, Ioannis Gkioulekas, Todd Zickler, Edward Adelson, and Kavita Bala. 2014. Looking against the light: How perception of translucency depends on lighting direction. *Journal of vision* 14, 3 (2014), 17–17.
- Bei Xiao, Shuang Zhao, Ioannis Gkioulekas, Wenyan Bi, and Kavita Bala. 2020. Effect of geometric sharpness on translucent material perception. *Journal of vision* 20, 7 (2020), 10–10.
- Wufeng Xue, Lei Zhang, Xuanqin Mou, and Alan C Bovik. 2013. Gradient magnitude similarity deviation: A highly efficient perceptual image quality index. *IEEE Transactions on Image Processing* 23, 2 (2013), 684–695.
- Cheng Zhang, Lifan Wu, Changxi Zheng, Ioannis Gkioulekas, Ravi Ramamoorthi, and Shuang Zhao. 2019. A differential theory of radiative transfer. ACM Transactions on Graphics (TOG) 38, 6 (Nov. 2019), 227:1–227:16. https://doi.org/10.1145/3355089. 3356522
- Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. 2011. FSIM: A feature similarity index for image quality assessment. *IEEE transactions on Image Processing* 20, 8 (2011), 2378–2386.
- Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. IEEE*

Conf. Comp. Vision & Pat. Rec. (CVPR). IEEE, New York, NY, USA, 586-595.

- Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. 2016a. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging* 3, 1 (2016), 47–57.
- Shuang Zhao, Ravi Ramamoorthi, and Kavita Bala. 2014. High-order similarity relations in radiative transfer. ACM Transactions on Graphics (TOG) 33, 4 (2014), 1–12.
- Shuang Zhao, Lifan Wu, Frédo Durand, and Ravi Ramamoorthi. 2016b. Downsampling scattering parameters for rendering anisotropic media. ACM Transactions on Graphics (TOG) 35, 6 (2016), 1–11.
- Quan Zheng, Vahid Babaei, Gordon Wetzstein, Hans-Peter Seidel, Matthias Zwicker, and Gurprit Singh. 2020. Neural Light Field 3D Printing. ACM Trans. Graph. 39, 6, Article 207 (Nov. 2020), 12 pages. https://doi.org/10.1145/3414685.3417879