Geometry-Aware Scattering Compensation for 3D Printing

DENIS SUMIN*, Max Planck Institute for Informatics, Germany
TOBIAS RITTIG*, Charles University, Czech Republic
VAHID BABAEI, Max Planck Institute for Informatics, Germany
THOMAS NINDEL, Charles University, Czech Republic
ALEXANDER WILKIE, Charles University, Czech Republic
PIOTR DIDYK, Università della Svizzera italiana, Switzerland
BERND BICKEL, IST Austria, Austria
JAROSLAV KŘIVÁNEK, Charles University, Czech Republic
KAROL MYSZKOWSKI, Max Planck Institute for Informatics, Germany
TIM WEYRICH, University College London, United Kingdom

Fig. 1. A demonstration of 3D printouts obtained with our method. The Earth diameter is 5 cm.

Commercially available full-color 3D printing allows for detailed control of material deposition in a volume, but an exact reproduction of a target surface appearance is hampered by the strong subsurface scattering that causes nontrivial volumetric cross-talk at the print surface. Previous work showed how an iterative optimization scheme based on accumulating absorptive materials at the surface can be used to find a volumetric distribution of print materials that closely approximates a given target appearance.

In this work, we first revisit the assumption that pushing the absorptive materials to the surface results in minimal volumetric cross-talk. We design a full-fledged optimization on a small domain for this task and confirm this previously reported heuristic. Then, we extend the above approach that is critically limited to color reproduction on planar surfaces, to arbitrary 3D shapes. Our method enables high-fidelity color texture reproduction on 3D prints by effectively compensating for internal light scattering within arbitrarily shaped objects. In addition, we propose a content-aware gamut mapping that significantly improves color reproduction for the pathological case of thin geometric features. Using a wide range of sample objects with complex textures and geometries, we demonstrate color reproduction whose fidelity is superior to state-of-the-art drivers for color 3D printers.

CCS Concepts: • **Computing methodologies** → **Reflectance modeling**; **Volumetric models**; • **Applied computing** → **Computer-aided manufacturing**.

Additional Key Words and Phrases: computational fabrication, appearance reproduction, appearance enhancement, sub-surface light transport, volumetric optimization, gradient rendering

---

*Denis Sumin and Tobias Rittig share the first authorship of this work.

Authors' addresses: Denis Sumin, Max Planck Institute for Informatics, Saarbrücken, Germany, denis.sumin@mpi-inf.mpg.de; Tobias Rittig, Charles University, Prague, Czech Republic, tobias@cgg.mff.cuni.cz; Vahid Babaei, Max Planck Institute for Informatics, Saarbrücken, Germany; Thomas Nindel, Charles University, Prague, Czech Republic; Alexander Wilkie, Charles University, Prague, Czech Republic; Piotr Didyk, Università della Svizzera italiana, Lugano, Switzerland; Bernd Bickel, IST Austria, Maria Gugging, Austria; Jaroslav Křivánek, Charles University, Prague, Czech Republic; Karol Myszkowski, Max Planck Institute for Informatics, Saarbrücken, Germany; Tim Weyrich, University College London, London, United Kingdom.

1 INTRODUCTION

High-resolution, multi-material 3D printers can deposit a wide range of materials next to each other at a resolution of a few microns

[Sitthi-Amorn et al. 2015; Stratasys 2016]. Not only do these capabilities enable fabrication of custom, highly detailed, and fully functional 3D objects, but they also have tremendous potential for printing objects in full color to reproduce complex surface appearance (Figure 1). For high-resolution full-color output, inkjet 3D printing with ultraviolet (UV) curable materials [Mimaki 2017; Stratasys 2016] has recently become the method of choice. The technology uses materials made of color pigments dispersed in a photopolymer medium. The critical problem of the current materials is their significant optical scattering which leads to a blurry look of printed objects. The effect goes well beyond the familiar "dot gain" observed in 2D printing [Hersch and Crété 2005], or thin-layered 3D printing [Babaei et al. 2017], for which direct means of compensation can be constructed. The above problem makes the reproduction of high-frequency texture details extremely challenging.

Solving this issue involves finding an optimal volumetric distribution of printing materials based on a target appearance and the material scattering properties. There are two main challenges in providing an effective and efficient solution to this problem. First, the vast parameter space made of millions of voxels, and the non-trivial relationship between material arrangement and light transport within the printed object pose significant computational challenges [Chen et al. 2013]. Second, the exact appearance reproduction, especially of fine geometric features, often turns out to be impossible in the presence of significant scattering. Although addressing the above challenges is critical for faithful appearance reproduction using new multi-material 3D printers, none of the existing techniques account for them simultaneously. They either ignore the effects of volumetric scattering in the printing materials [Brunton et al. 2015] or are incapable of reproducing complex geometry [Elek et al. 2017].

In this work, we propose a new technique for optimizing color prints that is both computationally efficient and considers volumetric scattering within general 3D shapes. Our method is informed by a series of experiments that, for the first time, study the nature of *optimal* material distributions at the example of small test cases, using an accurate, full-volume conjugate-gradient optimization over a Monte-Carlo-simulated voxel volume. One finding is that these solutions exhibit a trend consistent with a previously proposed heuristic by Elek et al. [2017] that steers higher-absorbing material assignments toward the object surface.

Subsequently, we base our technique on that heuristic, applying it to the general 3D setting in a new iterative space traversal that relies on a custom view-independent Monte Carlo light transport simulation and accounts for arbitrarily shaped objects. This ensures sharp texture reproduction – within physical limits – even around thin geometric features, where a limited material volume is available for optimization.

Our experimentation with the full-volume optimization also allowed us to derive a new geometry-aware color gamut. This is a crucial component of our technique since it provides a new way of dealing with the dependency of the achievable color gamut on the local geometry in the presence of material translucency. The mapping enables reliable handling of the appearance in thin geometry areas and provides plausible color reproduction even where physics limits the achievable gamut. We compare our method to state-of-the-art

techniques and demonstrate significant fidelity improvements. Our key contributions include:

- investigation of the solution space and validation of a previous material assignment heuristic using a full-volume conjugate-gradient optimization;
- design of a full-3D print-preparation pipeline that uses this heuristic for sharp, scattering-compensated texture reproduction on general 3D shapes;
- geometry-aware color gamut mapping where physical limits occur.

After discussing related work, we present our optimization-based study in Section 3 that informs our method presented in Sections 5 and 6.

2 RELATED WORK
Our work falls into the emerging field of appearance fabrication.

Reflectance fabrication. Earlier works aimed at fabricating custom surface reflectance using different hardware setups. Weyrich et al. [2009] designed microgeometries on a metallic surface to fabricate objects with predefined highlights. Matusik et al. [2009] created prints with spatially-varying bidirectional reflectance distribution function (BRDF) using a 2D printer capable of printing with diffuse and metallic inks. Lan et al. [2013] used a bi-scale approach to fabricate anisotropic reflectances: they combined the appearance of a small-scale height field, manufactured using a 3D printer, with material BRDFs, printed using a flatbed printer. Our work forgoes accurate surface reflectance control and instead focuses on high-quality color and texture reproduction.

Color 3D printing. Closely related to our work are the recent efforts on color reproduction [Babaei et al. 2017; Brunton et al. 2015; Elek et al. 2017] using UV curable inkjet 3D printing. Although various color 3D printing techniques exist – using technologies such as paper lamination, powder-binder or fused filament fabrication – the inkjet technology gives arguably the highest output quality [Sitthi-Amorn et al. 2015; Stratasys 2016]. Using this technology, Brunton et al. [2015] extended error diffusion halftoning to 3D objects, while relying on a conventional color prediction model from 2D printing. In a more recent work, Brunton et al. [2018] mixed CMYKW inks with a clear material to fabricate spatially-varying translucency. The contoning technique of Babaei et al. [2017] mixes inks through layering, thereby avoiding halftoning and its artifacts. Their color reproduction relies on an absorption-only color prediction model, justified with the use of non-scattering materials. Shi et al. [2018] extended this approach to spectral color reproduction, using a deep neural network for the contone stack prediction.

The above works ignore the important effects of lateral light scattering within the object, which induce unwanted color bleeding and texture blurring. Instead, they treat each small patch on the surface independently of its neighborhood. Elek et al. [2017] specifically targeted this problem and proposed a color reproduction method that preserves texture detail by optimizing the full volumetric material arrangement so as to match the target surface appearance. Using an accurate volumetric Monte Carlo appearance prediction, they have achieved important fidelity improvements, albeit only for

axis-aligned slab geometry. In our work, we validate their heuristic approach against a conjugate-gradient-based optimization. We then use it as a basis of our solution for general 3D geometries.

Translucency fabrication. Our current effort bears similarity to work on predefined translucency fabrication. Papas et al. [2013] determined pigment mixtures to reproduce homogeneous materials with a given color and translucency using silicon casts. Our work addresses a different problem: surface texture color reproduction for 3D printers. Dong et al. [2010] and Hašan et al. [2010] fabricated heterogeneous subsurface scattering appearance by locally optimizing material stacks. Dong et al. [2010] furthermore employed inverse diffusion [Wang et al. 2008] to account for the interaction between different neighboring layers. In contrast to these works that reproduce scattering profiles explicitly, our method considers subsurface scattering implicitly by recreating the diffuse input color map while taking into account translucency of the printing materials.

Precompensation. Our work is broadly related to the compensation techniques used in 2D printing for counterbalancing the dot gain [Rogers 1997]. The current best practice there is to account for the mechanical dot gain by adapting halftone dot areas [Hersch and Crété 2005]. Moreover, the optical dot gain is taken into account using the Yule-Nielsen factor [Ruckdeschel and Hauser 1978]. These empirical models are practical but limited to a simple ink-on-paper setup and uniform-color halftones.

In the context of 3D printing, some works attempted to counteract the dot gain effects. Cignoni et al. [2008] adapted the color depending on the difference between the appearance with and without subsurface scattering. Babaei et al. [2017] used a similar approach except that they measured the difference and used it for deconvolution of the target texture. Our pipeline, in contrast, predicts the exact light transport in a heterogeneous medium with non-planar geometry. Based on this prediction, the volumetric material arrangement is being refined.

Related to our thin feature treatment are works that enhance the shape to recover high-frequency geometric features which would otherwise be lost during the printing process [Herholz et al. 2017; Pintus et al. 2010]. While the geometry of our thin features is fully printable, it is *optically* too thin to reproduce arbitrary colors (consider a thin wall).

Forward and inverse volume rendering. Numerous approximate methods to simulate scattering in optically dense materials have been developed [Christensen 2015; D'Eon and Irving 2011; Donner et al. 2008; Jensen et al. 2001; Wang et al. 2008], their common deficiency being limited accuracy for non-planar geometries with heterogeneous scattering parameters – the very setup targeted by our method. Hašan et al. [2013] took advantage of light transport convexity for interactive re-rendering of arbitrarily shaped volumes. Their method only supports single-scattering albedo variations, while our printing materials additionally feature varying volume density. For this reason, our forward solution relies on full Monte Carlo simulation, an approach recently adopted also in the visual effects industry [Burley et al. 2018].

Our material arrangement optimization is essentially an inverse volume rendering problem. The approach has previously been used

to measure the scattering properties of homogeneous volumes [Gkioulekas et al. 2013]. Zhao et al. [2016] optimized the scattering parameters of a downsampled volume to match the appearance of an original high-resolution dataset. Finally, Gkioulekas et al. [2016] used time-resolved inverse rendering under structured lighting to recover spatially-varying scattering properties of a heterogeneous medium. A part of their solution is a full optimization similar to the one we use to validate our heuristic approach.

3 PROBLEM EXPLORATION

The space of possible voxel color assignments within a 3D-printed object is vast, forcing any practical solution for material assignment in print preparation to make use of heuristics and approximations [Babaei et al. 2017; Brunton et al. 2015; Elek et al. 2017]. Before developing our own heuristic, however, we would like to learn about the general nature of *optimal* solutions. To that end, we conduct a full-volume optimization over all voxel assignments, for small test cases where finding such an optimal solution is feasible. Of particular interest are cases with strong texture gradients and competing targets on opposing surfaces. Subsequent Sections 5 and 6 then develop a robust method that builds upon the insights gained from these fully-optimized small-scale solutions.

3.1 Full-volume Optimization

Our optimization operates on the continuous space of material concentrations at each voxel, aiming to find a (near) optimal material assignment that meets a target surface color specification for a small test geometry. Even for small cases, however, a full enumeration of possibilities is prohibitive. We hence chose a conjugate-gradient optimization to find a (potentially local) optimum. To hedge against non-convexity of the solution space, we run this optimization for various different initializations; robustness and generality are tested through different target specifications, material thicknesses, and error metrics.

Testbed. At its core, the setup consists of a voxel volume, representing a slab of finite, parametric thickness and practically infinite lateral extent (i.e., big enough to avoid undesirable light transport from the sides). As depicted in Figure 2a, it is constructed from a
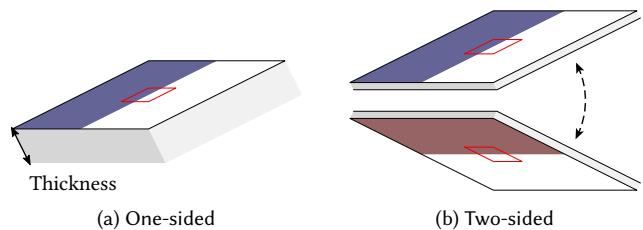

Fig. 2. Experimental Setup: One (a) or two (b) target textures are arranged into a centered region of interest (red square) on the front (or back) of a slab of parametric thickness. Edge-padding enlarges the volume to eliminate in-scattering of external lateral illumination.

texture patch centered at the region of interest and edge padding ("repeat" boundary extension) for expanding the volume laterally.

In order to keep the study general and independent of a particular set of printer materials, our testbed does not employ halftoning with its limiting discretization to obtain a heterogeneous medium. Instead, the linear RGB color at each voxel with applied Inverse Albedo Mapping (from [Elek et al. 2017, Appendix A.1]) is taken as the scattering albedo; density is set to a fixed uniform value comparable to existing printer materials.

The volume is assumed to have a smooth dielectric boundary ($\eta = 1.5$) and is illuminated using constant omnidirectional lighting. We modified a volume path tracer in Mitsuba [Jakob 2010] to compute alongside a radiance estimate also the partial derivatives of the latter w.r.t. each voxel's scattering albedo (i.e., the optimization variable). That renderer is then used in the inner loop of a conjugate-gradient least-squares solver [Branch et al. 1999] (in the implementation of [SciPy 2008]) to find material assignments that minimize the difference between predicted and target surface appearance.

Additionally to the optimization of planar scattering, this setup can be employed in a two-sided mode, in which it is able to optimize for two surfaces on opposing sides concurrently (Figure 2b). By reducing the thickness of the slab, these two surfaces are less and less independent, but rather influence each other due to the light traveling through the object.

Findings. In order to characterize the solution space for scattering compensation in 3D-print preparation, we ran a broad study over different input parameters, such as volume initializations (random, white, target extrusion, deep embedded confusion shape), slab thicknesses (in the range of [0.5, 10] mm) and error metrics (RMS in linear RGB, CIE dE76 (CIELAB)).

The resolutions of the target textures and the actually optimized volume are set to 16×16 pixels/voxels in the tangential plane, and we experimented with 8, 16, and 32 depth layers, respectively. At that resolution, each iteration of the optimizer runs in 10 seconds, with convergence after typically 5 to 30 minutes.

From our experiments – see a collection of generated reports in the supplemental – we gathered the following findings.

(1) As expected, perfect separation of front and back signal is not possible, given the materials' mean free path.
(2) Voxels with high absorption occur closer to the surface. This effect is more pronounced the thinner the material slab is (see Figure 3, rows 2–3) but is still not able to compensate for the increased cross-talk between back and front.
(3) Altering the metric for which we optimize does affect the internal distribution of the solution, but (a) the trend of higher absorption close to the surface persists, and (b) the visual differences in the corresponding outcomes are relatively small.
(4) Initialization influences the solution structure, but resulting appearances exhibit only negligible visual differences.

A useful finding is that (2) qualitatively confirms a hypothesis underlying the work by [Elek et al. 2017], who postulate that forcing strong absorbers closer to the surface leads to better texture reproduction in the presence of strong subsurface scattering. From (3) we further conclude that regardless of the error metric chosen, the outcome is dominated by the physical limitations. In the remainder, we hence use L2 in linear RGB. Nevertheless, an interesting difference remains that the use of CIE dE76, which puts more weight on color
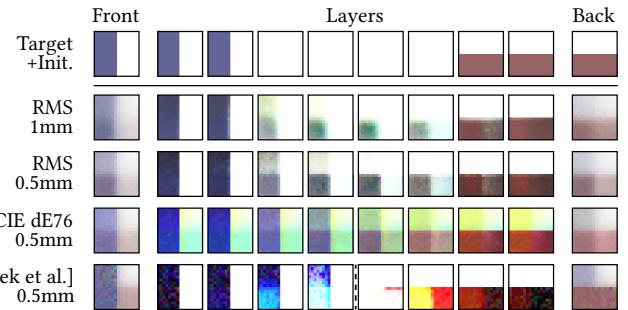

Fig. 3. Results of a conjugate-gradient optimization on volume assignments for a thin, two-sided slab. The columns show front and back predictions with the corresponding volume as horizontally scattered layers in between. Each row names the optimized difference metric and slab thickness.

accuracy, leads to the deposition of complementary colors on the opposing side (Figure 3, row 4), albeit the resulting visual difference is minute. Lastly, (4) is suggestive of our results being very close to the global optimum.

3.2 Heuristic Optimization

Larger volumes will no longer be amenable to general optimization, due to the substantial overhead of dense partial-derivative computation. In order to scale to larger problem sizes, we require a more heuristic approach that approximates these types of solutions at a lower computational cost.

To our knowledge, the existing heuristic whose volumetric assignments come closest to what we observed in the full-volume optimization, specifically in how darker materials occur near the surface, is the work by [Elek et al. 2017]. In the remainder, we conducted additional experiments to directly compare their method to the general optimization.

As their approach is limited to planar, thick objects, without any effects from an opposite surface, we ran a global optimization under this exact geometric assumption. In direct comparison (Figure 4), the solution of the general optimization resembles Elek et al.'s solution, except that they seem to more aggressively push higher saturations toward the surface producing slightly sharper contrast than the general optimization; however, this comes at the cost of a 27% higher RMS Error (General optimization: 0.120; Elek et al.: 0.152).

Now, explore a straight-forward adaptation to two-sided cases, where we alternate two parallel iteration loops of their algorithm, each modifying voxels only half-way into the volume from opposite sides, but using Monte Carlo predictions that simulate the thin slab as a whole. As visible in the last row of Figure 3, their heuristic does not have a notion of opposing surfaces. It will simply ignore their demands resulting in sharper edges and better uniformity but at the cost of more visible cross-talk.

Overall, the qualitative agreement of the solutions by Elek et al.'s method made us choose their algorithm as a basis of an extension to general 3D shapes, and for situations with opposing surfaces near thin features.
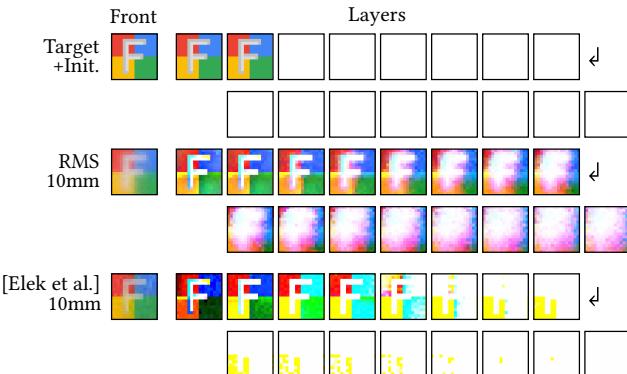
Fig. 4. Result comparison of a conjugate-gradient optimization on volume assignments with the heuristical approach [Elek et al. 2017] for a thick, one-sided slab (10 mm of white with the top 1 mm optimized). The columns show the prediction with the corresponding volume as horizontally scattered layers next to it.

4 PLANAR SCATTERING COMPENSATION

This section recapitulates the planar scattering compensation algorithm by Elek et al. [2017], which we will extend. They propose an iterative scheme that alternatingly predicts a current material assignment's appearance, followed by a heuristic update step that aims at bringing that appearance closer to the target.

This update step capitalizes on the aforementioned heuristic that, for crisp texture reproduction, the more absorptive (i.e., darker) materials should be deposited nearer to the surface, while lighter materials may be placed at greater depth. This asymmetry becomes intuitive when considering that light absorbed by darker materials is irretrievably lost, regardless of the depth at which it occurs, while light traveling through lighter materials can still be modulated by darker materials nearer to the surface to create fine-scale control over the surface appearance.

Accordingly, the optimization's update step treats the material assignment as a darkening process that prefers near-surface depositing of highly-absorbing materials over deeper locations. In each optimization step, negative parts of the residual, where the current solution is predicted to be lighter than the target appearance, are driven into the solution by gradual darkening of the voxel assignment. The aim is to deposit the absorptive material as close to the surface as possible and only propagating darkening to deeper levels once all voxels nearer to the surface exhausted their maximum saturation. In order to avoid excessive darkening that can no longer be compensated by lighter materials nearby (overshooting), the optimization proceeds carefully, each time applying only a fraction (50%) of the residual.

The residual propagation takes place strictly along voxel columns underneath each surface texel, ignoring neighboring content; only once the Monte Carlo prediction step is run, lateral scattering effects become apparent and will be countered by the next update step. This allows for an efficient update and remains stable, as the bulk of the energy within a surface points' subsurface scattering kernel always lies close to that point.

The method creates high-quality results, but is inherently limited to planar surfaces on bulk material, due to its reliance on axis-parallel voxel columns under each pixel and through the lack of a mechanism to take color cross-talk near thin features into account.

5 GENERAL GEOMETRY SCATTERING COMPENSATION

In this section, we build a full general-geometry pipeline that can compensate for sub-surface scattering in 3D prints. Figure 5 gives and overview of the whole process, from the input appearance to the final printout. First, we convert ① the target shape with a diffuse texture to a voxel representation as described in Section 5.1. Our initial solution is created in step ②. We iterate over candidate solutions using our refinement loop, consisting of halftoning ③ (Section 5.3), prediction ④ (Section 5.4) and volume refinement ⑤ (Section 5.5). Finally, the halftoned volume is ready for fabrication ⑥.

5.1 Data Preparation

Using OpenVDB [Museth 2013], we convert the input 3D mesh into a distance transform on an isotropic voxel grid (step ①). As the resolution along the printer's three axes usually differs (600×300×940 DPI in our case), we choose to work with the coarsest of the three. Later, during halftoning (Section 5.3), we up-sample to the printer's native resolution. The coarser solution grid allows for significantly faster processing; conversely, optimization at the full resolution would have little benefit, as the need for halftoning reduces the effectively available printer resolution anyway.

We classify voxels as *interior*, *surface*, or *exterior*. Surface voxels encode the corresponding surface normal and gamut-mapped, target RGB color. We perform the gamut mapping in CIELab color space using a state-of-the-art algorithm [Rossier 2013]. Note that this is a conventional gamut mapping that ensures uniform RGB colors are within the printer gamut [Elek et al. 2017], different from our content-aware gamut mapping introduced in Section 6. The solution for the volumetric material arrangement is stored as linear RGB values representing the desired absorption at each non-exterior voxel of the coarse grid.

5.2 Interior-to-Surface Points Mapping

In planar scattering compensation (Section 4), each voxel layer corresponds to a distinct depth within the material, while columns group all voxels underneath a given surface point. For generally shaped 3D prints, there is no clear definition of voxel layers and columns. We propose three distinct changes to account for surface curvature and geometric thickness. First, we replace the notion of layers by partitioning the voxel object into *iso-shells* of equal (after rounding) distance to the surface. Second, lacking a precise definition of voxel columns in curved regions, we store for each non-exterior voxel its nearest surface voxel to associate voxels with a surface point.

While this mapping is sufficient in thick geometric cases, it becomes misleading for thin geometries where a considerable amount of light enters through adjacent surfaces (Section 3.1, Finding 1). Voxels thus affect the appearance of several surfaces, not only the closest one, which necessitates an appropriate representation for the refinement step (Section 5.5).
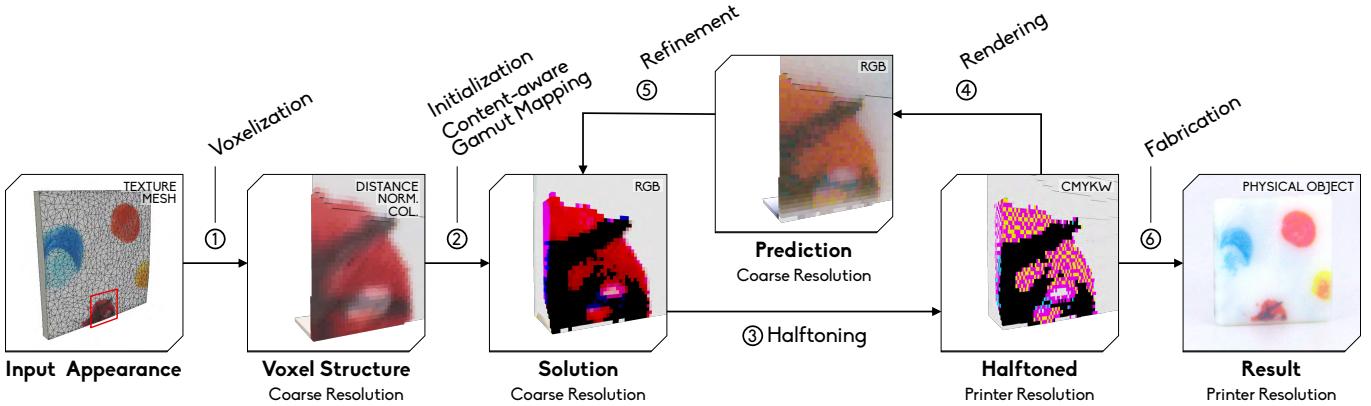
Fig. 5. Our pipeline takes a target in the form of a textured triangle mesh. It voxelizes it ① at a uniform resolution with additional data stored on the surface. The inner loop iteratively optimizes an RGB volume by predicting the appearance ④ and propagating the residual error into the volume ⑤. Finally, we output discrete material assignments readable by the 3D printer ⑥.

Accordingly, our third adjustment is a conceptual augmentation of the voxel-to-surface correspondence function (implicit in [Elek et al. 2017]) to multiple connections. Motivated by the energy distribution of a subsurface scattering kernel, the connections are concentrated around the surface normal direction and sorted by distance to model the major influence.

In practice, we compute additional surface connections for each inner voxel in thin geometries as shown in Figure 6. Through uniformly shooting rays over the sphere against the input geometry, connections are initially discovered (a). Discretized to voxel indices, the candidates are weighted according to their deviation from the surface's normal direction. We form clusters of direction vectors (b) and improve each cluster by shooting additional samples drawn from a von Mises-Fisher distribution fitted against the current best $\tau = 8$ candidates per cluster (c-d). By iteratively re-fitting the distributions, cluster directions can change and potentially merge (e). Typically after 8–10 iterations of 64 samples each, the algorithm converges to a list of surface connections that are aligned with the normal direction (f). Eventually, this list (sorted by distance and capped to $\lambda = 8$ items) contains only surface voxels whose directions are sufficiently separated in order to circumvent duplicate connections to the same surface.

5.3 Halftoning

Predicting the appearance of the solution at each refinement stage (and ultimately fabricating the model) requires both conversion into the operational grid of the printer and discretization into the material labels (CMYKW) as our target 3D printer can only place a single material at each voxel. We perform this as a single operation ③. For each point at the printer grid (higher resolution), we query the RGB values from the solution represented on the coarse grid using tri-linear interpolation. The RGB value is then converted to CMYKW tonal mixture using a static mapping model [Elek et al. 2017]. Finally, we perform a per-slice error-diffusion halftoning using a serpentine-order with the Floyd-Steinberg kernel [Floyd and Steinberg 1976]. Although more advanced 3D halftoning techniques are available [Brunton et al. 2015], for our examples, the difference was not visually significant.

5.4 Appearance Prediction

The material arrangement refinement ⑤ of our method relies on accurate prediction ④ of the surface appearance implied by the current solution. We employ Monte Carlo simulation for this purpose, using the scattering parameters of the print materials reported by Elek et al. [2017]. While time-consuming, this technique remains – to our knowledge – the only method capable of accurate predictions in the presence of complex geometric features (sharp edges, thin slabs, etc.), while also supporting heterogeneous media.
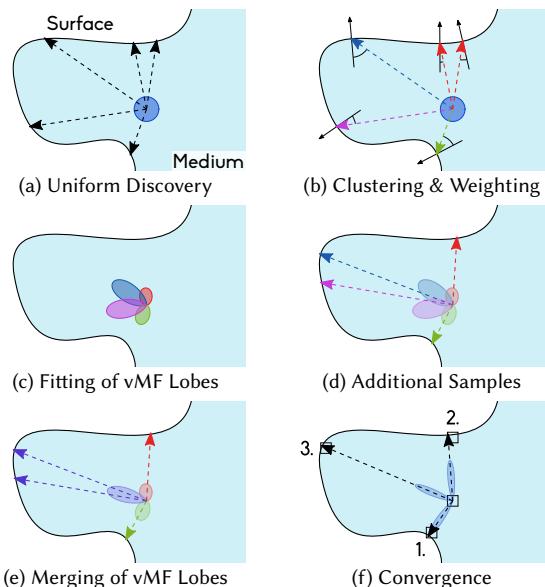

Fig. 6. Interior-to-Surface Mapping: Connections are initially discovered using uniform sampling (a) and further refined using fitted von Mises-Fisher (vMF) distributions (b-e) until convergence in surface normal direction (f). The result in (f) is sorted by distance and references discrete surface voxels.

The volume grid input to the prediction stage consists of one of the five measured material parameters per voxel [Elek et al. 2017] at the printer's native resolution. The lighting simulation sees the grid as block-shaped voxels, assuming a hard boundary between neighboring voxels, though in practice material mixing and mechanical dot gain may occur at voxel boundaries.

Light transport model and lighting. The (virtual) lighting setup greatly influences the object appearance, as the print materials show a high degree of translucency. Optimizing the printout for specific lighting is feasible, however, it would reduce its generality. For this reason, we assume spatially and directionally constant, white illumination, factoring out all possible influences of the environment (such as self-shadowing or inter-surface reflections) from the optimization procedure. In fact, the prediction step is designed to only consider the subsurface part of light transport, which is the primary effect our method addresses. Specifically, our volumetric path tracer (implemented in Mitsuba [Jakob 2010]) stops tracing a path once it exits the medium through the surface into the air, and each such path brings unit radiance modulated by the Fresnel transmittance for a smooth dielectric boundary ($\eta = 1.5$).

Sensor. Our goal is to match the appearance over the entire object surface, so predictions obtained with standard perspective or orthographic cameras would be insufficient. Performing the prediction directly in the UV texture space would require special consideration of texture distortion and discontinuities. Instead, we take advantage of our volumetric representation with the surface normal assigned to each surface voxel. This data is input to a custom sensor in Mitsuba, which renders for each 'pixel' the outgoing radiance of one surface voxel. As depicted in Figure 7, the 'viewing' direction for each voxel is aligned with its surface normal so that the transmission at the dielectric boundary is the same for all voxels.

While the light transport simulation runs on the printer-resolution grid, the predicted outgoing radiance is recorded at a set of surface samples, corresponding to the centers of the surface voxels of the coarser grid. This makes the prediction output ready for the subsequent stages of our pipeline, which work at the coarser resolution.
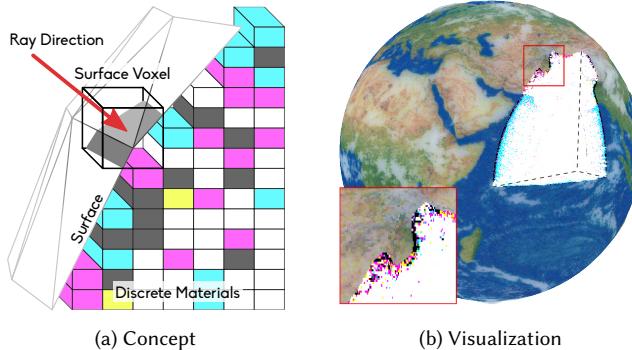

(a) Concept   (b) Visualization

Fig. 7. Rendering setup: For each surface voxel (in the solution resolution), we render point samples along the surface's normal direction (red arrow in (a)) and trace rays into the discretely heterogeneous medium (printer resolution). The visualization in (b) shows a typical surface prediction (stored in voxels) with the underlying halftoned solution as a cubic cross-section.
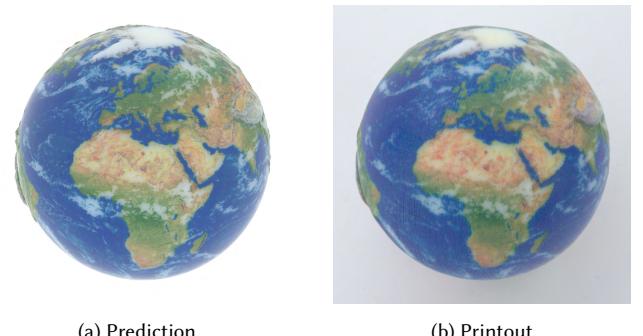

(a) Prediction   (b) Printout

Fig. 8. Prediction accuracy: (a) shows a *preview* prediction (perspective sensor, spherical illumination) of the Earth model. (b) is a photograph of the corresponding printout.

Preview setup. For verification and demonstration purposes, the halftoned solution can be rendered with a conventional perspective sensor and omnidirectional illumination. An image obtained in such setup is compared in Figure 8 to the printout showing the quality of our prediction system. Figure 12 additionally offers a comparison for thin geometry (planar 0.5 mm slab). The *preview mode* is used in further figures and is explicitly marked with the term *rendered*.

5.5 Material Arrangement Refinement

Each iteration of our optimization loop (Figure 5) consists of rendering ④ the current (halftoned) solution and a refinement step ⑤, in which we propagate residual error into the volume. We initialize the solution by extruding the surface color into an initially fully white object up to a depth of $d = 0.1$ mm, by assigning the target color of its closest surface point to each interior voxel in the depth range $[0, d]$. The thickness parameter d influences the overall convergence speed, as it allows our method to skip a few early iterations if it is chosen well: but ultimately, it does not significantly affect the sharpness of the result.

Update step. The rendering step provides an appearance prediction per surface point, from which a residual error compared to the target color is calculated. Positive residual values in a color channel imply that the current solution is too dark and needs lightening, while negative values indicate that darkening is required.

For each surface point, the refinement step distributes this residual into the set of voxels that correspond to this surface point (Section 5.2). Most of it is deposited into voxels which reference it as the *closest* surface point, while leftover lightening is additionally also propagated to voxels which 'belong' to a different side of the object (i.e. that only hold a higher-order reference to this surface point).

We traverse all iso-distance layers, up to a maximum depth beyond which no noticeable light transport can occur – which in the case of our printer materials is 3 mm (corresponding to 18, 27, 72 mean-free-path lengths for the R, G, B channels of the white material respectively). In the refinement procedure (Algorithm 1), we first traverse the volume from the deepest iso-layers up to the surface and disperse any *positive* residual components, *lightening* the RGB values within the grid. Next, and similar to the planar compensation approach (Section 4), a *darkening* pass processes the volume from

---

ALGORITHM 1: Refinement (update step) ⑤

Input:

X: current solution volume

T: target-specification surface voxels

P: current appearance-prediction surface voxels

Result:

X: updated solution volume

begin

    // Calculate positive and negative differences from target

    // $D[voxel, c]$ returns the c-th order voxel-to-surface correspondence

    $D = T - P$; $D^+ = \max(D, 0)$; $D^- = \min(D, 0)$

    // Lightening and Darkening passes

    Lighten(D^+, 0)

    Darken(D^-)

    // Pass over secondary voxel-to-surface correspondences to disperse the

      residual lightening to the rest of the volume; we use $\lambda = 8$

    for $correspondence = 1..\lambda$ **do**

      | Lighten(D^+, $correspondence$)

    end

end

procedure Lighten(D^+, $correspondence$):

    // Loop over iso-layers L_z, from the most distant one Z to the surface.

    for $z = Z..0$, $L_z \in X$ **do**

      // Loop over voxels belonging to the iso-layer.

      forall $v \in L_z$ **do**

        | $v \mathrel{+}= D^+[v, correspondence]$

        // Update of the positive-residual for a given surface point

          ensures (unless close to the surface) the compression

        if $v.distance \geq 0.25mm$ **then**

        | $D^+[v, correspondence] = \text{clamp}(v - 1, 0, 1)$

        end

        $v = \text{clamp}(v, 0, 1)$

      end

    end

end

procedure Darken(D^-):

    // Loop over iso-layers L_z, from the surface to the most distant one Z.

    for $z = 0..Z$, $L_z \in X$ **do**

      forall $v \in L_z$ **do**

        // Conservatively darken the voxel proportionally to c_p.

        // $c_a = 4.0$, $c_p = 0.5$ as in [Elek et al. 2017]

        $v \mathrel{+}= c_p \cdot D^-[v, 0]$

        $D^-[v, 0] = \text{clamp}(c_a \cdot v, -2, 0)$

        $v = \text{clamp}(v, 0, 1)$

      end

    end

end

---

the surface iso-layers to the deepest ones, to disperse any *negative* components. Last, we again run multiple sparse lightening passes; this time depositing leftover positive residual only to voxels with an additional reference (i.e. *not* closest) to a surface point.

These references (derived in Section 5.2) are only available for voxels in thin regions and are sorted by ascending distance which in turn means by descending influence. The last lightening pass enables 'negotiation' in cases when 'demands' of the two sides of a fine geometry are in conflict, i.e. one side should be darkened (in

one of the R, G, B channels) while the other side should be lightened. The approach ensures that absorbing voxels are placed only near the darker surface and, at the same time, allows for processing the volume as a whole as opposed to processing two halves of the volume separately.

Recovery from overshooting. Similar to [Elek et al. 2017], we limit the fraction of residual dispersed in each update step, and propagate darkening and lightening only if the current iso-layer voxel can no longer receive any more darkening or lightening.

Even so, situations may still occur where the update leads to excessive darkening for individual surface points (for example due to a concurrent darkening of an opposing surface). In order to recover in these cases, Elek et al. [2017] propose to uniformly lighten all voxels within the corresponding column whenever the residual is positive. However, such method of recovery disturbs the goal to minimize the number of absorbing voxels by placing them very close to the surface.

This problem is most severe when the target texture contains fine, dark details surrounded by larger areas of light material. Here, the dark surface voxels are back-lit by comparatively large amounts of in-scattered light. Any change of their concentration, such as darkening by changing the local absorption by a fixed amount, is amplified by the higher "back lighting" (absorption is multiplicative), increasing the risk of overshooting. In these cases, Elek et al.'s uniform lightening implicitly draws the bulk of the darkening deeper into the volume, increasing dark color bleeding into the light surrounding and decreasing the globally achievable contrast. This effect is demonstrated in Figure 9, where, comparing the *rendered* predictions, a decrease in contrast is visible and the loss of concentration at the surface is revealed in the slice through the halftoned solution.

In contrast, we rely on our new inside-out lightening pass described above to ensure that the darker voxels stay near the surface, mitigating potential blur. This *compressive* behavior is only stopped in the very proximity of the surface (0.25 mm in our system): shades of colors are better reproduced with 2 or 3 iso-layers than with a single iso-layer due to the small number of base materials (and, consequently, color combinations) in the subsequent Halftoning

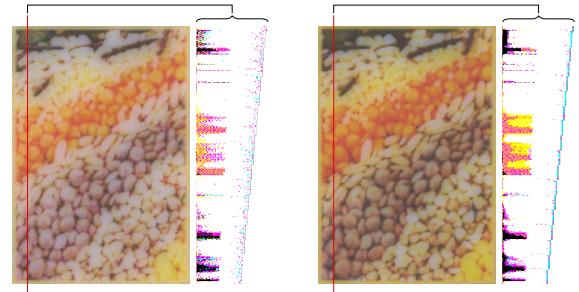(a) [Elek et al. 2017]   (b) Our Update Step

Fig. 9. Comparison of optimization strategies: The textured side (rendered) of planar slabs with varying thickness (3–4.3 mm) and uniform blue target color on the back side. Cross-sections show the arrangement of printing materials in the volume. Our strategy recovers well from over-darkening due to cross-talk and preserves absorption close to the surface.

step ③. We observe that the new update step is effective in eliminating overshoot artifacts, to the extent that we are able to use a 3-times more aggressive value for darkening factor c_a, which results in faster convergence.

6 CONTENT-AWARE GAMUT MAPPING

Smaller models often possess thin geometric features where one cannot assume enough volume for the printing materials to reproduce a range of colors. The colors on thin features are most severely affected by the colors on the opposite surface due to backlighting. This is a fundamental limitation of current color 3D printing technology and no solution comes without compromises. Our main insight for treating these cases is to adjust the color of input models to the capacity of geometry they rest on. This will reduce the risk of artifacts appearing due to insufficient volume and conflicting demands during our iterative refinement stage (Section 5.5). Note that the introduced content-aware gamut mapping is different from the conventional gamut mapping (Section 5.1) performed for all input colors. The latter is independent of the surrounding colors.

Babaei et al. [2017] pointed this problem out and offered a procedure in which they perform an adaptive gamut mapping where each color on the surface is mapped into the gamut afforded by the object thickness at that point. This approach, however, is limited, as it does not consider the actual colors in the vicinity. For a thin wall with nearly identical colors on both sides, it would drastically underestimate the practically achievable gamut, because it only considers half of the depth for each side. If colors on the two sides of a thin wall are contradicting, they would likely suppress each other.

We instead take into account both geometry and color *content* when remapping input colors in our preprocessing (step ② of our pipeline, see Figure 5). We take advantage of our full-volume conjugate-gradient optimization setup (Section 3.1) to investigate the range of achievable color pairs for different object thicknesses. Then, we use this data to process any pairs of conflicting colors and bring them into the gamut of realizable color combinations.

6.1 Gamut of Achievable Color Pairs

In this section, we describe a one-time precomputation of the content-aware gamut. It consists of two steps: (1) collection of achievable color pairs for different thicknesses to understand the boundaries of the gamut in the color space of our prediction system (RGB), and (2) building a dense set of realizable color combinations (again, for

different thicknesses) in a perceptual color space for performing the actual gamut mapping.

Collection of achievable RGB combinations. Using the two-sided setup (Figure 2b), we run the conjugate-gradient optimization for thin walls of 10 different thicknesses from 0.25 to 3 mm. On the two sides, we place all combinations of uniform RGB colors where each channel is set to one of three values (0.0, 0.5, 1.0); this gives us 27 colors (Figure 10a) and 351 color combinations. The final result of each optimization is a pair of colors which are both nearest to the target pair and realizable at a specific thickness (Figure 10b). We also include the intermediate results obtained from each optimization iteration (around 20) as each rendering is an achievable pair on its own. Including these intermediate pairs helps to obtain a larger set of achievable color pairs (around 4000 for each thickness).

We use a per-channel *mean-difference* representation to process the calculated data: each pair of colors on the two sides – RGB and $R'G'B'$ – is transformed into:

$$\left(\frac{(R + R')}{2}, |R - R'|\right); \left(\frac{(G + G')}{2}, |G - G'|\right); \left(\frac{(B + B')}{2}, |B - B'|\right)$$

This representation captures the following features. First, the cross-talk can be estimated in each channel (R, G, B) separately; we did not find any inter-channel dependencies. Second, the achievable *difference* values are dependent on the *mean* values; e.g., the brighter the colors, the smaller the achievable difference. Third, the achievable differences are symmetrical with regard to the two sides.

Building the gamut. In order to build a dense, discrete (content-aware) gamut we begin with computing the gamut boundary by finding convex hulls of 2D projections of the previously simulated color pairs. Figure 11 demonstrates the convex hulls of the collected data in the *mean-difference* representation for different wall thicknesses. Then, we tightly sample the CIELCh color space [Wyszecki and Stiles 1982] (collecting 163 million pairs) and verify if each *pair* of LCh colors is achievable at each of 10 thickness values. Our verification consists of checking if a given LCh pair is inside the boundary of the set of achievable colors obtained in the previous step. Given a thickness, for each LCh pair, we compute its equivalent RGB pair and check if all 2D projections of this pair (RGB on one side, $R'G'B'$ on another) are within the convex hulls. If the test is valid for all projections (R/R', G/G', B/B'), we mark the color pair in-gamut.
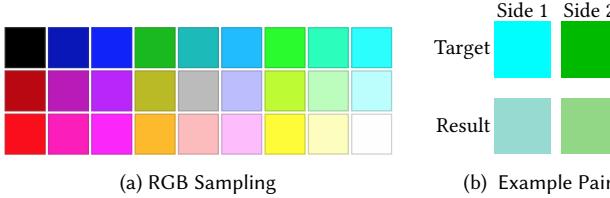

Fig. 10. (a) The palette of target colors used to build the *convex hull* of content-aware gamuts. (b) One sample assignment for a two-sided wall (Figure 2b, thickness: 0.5 mm) with a target color on either side. Using conjugate-gradient optimization, we transform the target pair to an achievable pair for each wall thickness.
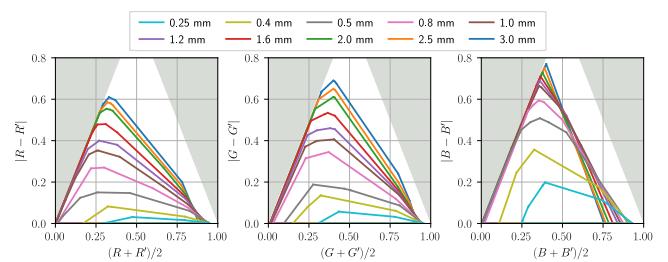

Fig. 11. Convex hulls of the achievable color pairs in the *mean-difference* representation for each channel and thickness of the object. Each convex hull represents around 4000 data points.

Fig. 12. Result of the content-aware gamut mapping demonstrated on planar slabs of different thicknesses (0.5, 1.0, 2.5 mm) with two images as target (a). (b) demonstrates the gamut-mapped target specification. The images on the gray background (c) are photographs of printouts as opposed to the rendered solutions in (d). Note the reduced cross-talk (i.e. the window under the arrow) and the reproduction of details and colors for the thinner cases.

6.2 Mapping

Upon having the *content-aware* gamut of achievable color pairs for a set of thicknesses, we – once for an object (Figure 5, step ②) – process the input colors depending on the nearby geometry and the close colors on the opposite side. This step ensures that conflicting colors on different sides of thin geometric features are reproducible. It is, in fact, an approximation to the solution which a full-volume conjugate-gradient optimization would produce if run on thin features, except the expensive optimization is replaced with gamut precomputation.

We experimentally observe that the backlit cross-talk is present when the object is thinner than 3 mm. Our content-aware gamut mapping, therefore, acts on parts of the model with thickness below 3 mm. Before assigning conflicting pairs, we smooth the surface colors using a Gaussian filter. This way each color receives collective feedback from the other side, as we relate each color on one side to only a single conflicting color on the other side of a thin feature.

We iterate over all voxels and collect pairs of conflicting surface voxels that are (1) on opposite sides (see Section 5.2) and (2) closest to each other. The opposite-side condition is checked by analyzing the mutual arrangement of the two surface voxels and the connecting inner voxel. For example, in the volume demonstrated in Figure 6f, surface points 1. and 2. are marked as *conflicting* if the distance between them is less than 3 mm.

If any pair of conflicting colors is outside the gamut of achievable color pairs (Section 6.1), it will undergo an adjustment. This includes a simple lookup in the densely-sampled space of achievable LCh color pairs and finding the closest point using the CIEDE94 distances. The result of this preprocessing is a set of modified colors at thin features which are fed to the subsequent stages of our pipeline (Figure 5).

Content-aware gamut mapping limitations. A major limitation of our method is that for each voxel we assign only a single conflicting color, lying on the opposite side. Although our smoothing step, indirectly, causes more than one voxels to influence the opposite side, one could argue that some neighboring voxels on the same side could have an effect as important as the opposite ones. In principle, one could incorporate more conflicts but should be careful that our method does not scale well with more colors as the number of projections grows unwieldy. Apart from this limitation, our nearest neighbor search is essentially a *gamut clipping* which is a less desirable category of gamut mapping techniques [Morovic and Luo 2001]. Gamut clipping preserves the saturation but may lose image details. The more acceptable *gamut compression* category is however computationally intense in our 6D setup.

(a) Target   (b) No Content-aware   (c) Content-aware
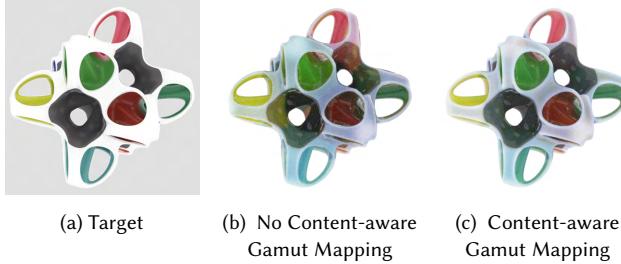   Gamut Mapping   Gamut Mapping

Fig. 13. The effect of our content-aware gamut mapping demonstrated on an arbitrarily shaped object (rendered). The thickness of the geometry is ≈ 1 mm. Our proposed technique reduces cross-talk and improves the spatial consistency of the object.

Content-aware gamut mapping results. Figure 12 illustrates the effect of content-aware gamut mapping on thin planar slabs. We place two different textures (a) on the two sides of a slab. The bottom row shows the results we obtain using our normal procedure (Section 5), where the target images have gone only under a conventional gamut mapping. The results of both sides of the unprocessed targets exhibit artifacts at the thickness of 0.5 mm, along with hue shifts and loss of details. See, for example, how the window (marked with arrows) is visible on the opposite side of the printout or the objectionable green tint on the pillow in van Gogh's *Bedroom in Arles*. At the thickness of 1.0 mm, we see the gradual decrease of the cross-talk, and at 2.5 mm it is almost not visible. The middle row shows the results when using the gamut-mapped target specifications (b). As we can see, they not only achieve a much more consistent match when compared against their own target textures but also show good consistency with the original textures. Additionally, the images on the gray background (c) demonstrate the printed 0.5-mm slabs. They are consistent with the renderings (d) and also demonstrate the achieved improvement.

Figure 13 demonstrates the effect of content-aware gamut mapping on a generally-shaped model. This object has a height of 3 cm and features approximately 1-mm thick 'walls'. The cross-talk is significantly reduced when using the content-aware gamut mapping. The proposed solution is more consistent with the target specification: the white parts of the model preserve the neutral color, the change of color on the opposite side of geometry is not visible.

7 RESULTS

We fabricated the presented results using a Stratasys J750 PolyJet 3D printer and the Vero Rigid Opaque materials family. The printouts were cleaned using a grade-1000 sandpaper and sprayed with a transparent lacquer. We avoid scratching the absorbing voxels by adding a ≈0.12 mm shell of transparent material around the object. The pictures of the printouts were taken using a Canon EOS 700D camera with EF-S 18–135 lens set to 135 mm. We used two 55W, 5500 K fluorescent lamps and X-Rite ColorChecker Passport to set the white balance and the luminance level.

The refinement and appearance prediction were performed on isotropic voxel grids with resolution 300 DPI. The Monte Carlo simulation for appearance prediction was performed on a CPU cluster consisting of 100 quad-core Intel Xeon E5620 CPUs. It takes

around 5 minutes to predict the appearance for a 4 cm object (i.e. red vase from Figure 14).

Target reproduction evaluation. The main results are presented in Figure 14. For comparison, we fabricate the exact same models using our pipeline and two additional solutions: the built-in printer software (GrabCAD) and Cuttlefish, the commercially available software package that, to our knowledge, represents the approach described in [Brunton et al. 2015]. Our results are presented in the rightmost column.

In general, results produced using our method show significantly higher reproduction fidelity in terms of both color and spatial details. This is expected as the two other methods are agnostic to the mutual volumetric light transport between nearby surface points. The Cuttlefish printouts seem to be slightly color shifted, which may be explained by inter-machine variability in calibration data. Our thin results, especially the last row, show dramatic improvement when compared to the other methods: they are closer to the source image color-wise and suffer from much less backlit crosstalk.

One peculiar case is the color reproduction of the cheetah model. None of the methods is able to reproduce the orange segments of the skin pattern. We investigated this case and found out that both black and orange segments are reproducible when printed uniformly, isolated from each other. We then run a full-volume optimization including these two colors together and observe that even the full optimization is not able to reproduce them simultaneously. Although we are aware that this may be an extreme case (strong absorbing material next to a material with a large mean free path at high spatial resolution) where both our refinement and full-volume optimization fail to find a valid arrangement, it can plausibly show us the physical limits of this set of materials.

Figure 15 presents additional photos of the thin slabs printed with different methods (row 5 in the main results). Here, the prints are lying on a white table (rows 1–2) and on a black-white edge (row 3–4) in contrast to a vertical standing in the main-results figure. The absence of back-light changes the perceived appearance; the black-white edge demonstrates the extent of translucency we are dealing with.

7.1 Baseline Comparison to Remapped Planar Slabs

Lastly, we would like to validate whether a simpler extension of Elek et al. [2017] would have been possible. Intuitively, their method could also be generalized by optimizing a texture as-is in 2.5D and treating the resulting planar volume as a 3D texture that fills a 3D mesh. Such remapping would take into account the UV parametrization on the surface and the distance into the volume to select the correct texel on the planar surface and layer from the stack respectively. A comparison of this idea to our native 3D treatment of the problem is described in the following.

We conduct this baseline experiment to evaluate the texture sharpness and color reproduction on 3D surfaces achievable by both methods. The remapping approach has disadvantages when implemented naïvely and requires special care to avoid pitfalls such as texture stretch or a mismatch in light scattering distance before and after the mapping. Please refer to the supplemental for a detailed description of our implementation.

Fig. 14. Comparison of photographed printouts to the rendered (a) target models: (b) GrabCAD is the default printer software, (c) Cuttlefish is based on [Brunton et al. 2015], and (d) Our scattering- and crosstalk-compensated solution. The last model is a thin planar slab of 0.5 mm thickness with two different textures on its front and back sides.

(a) GrabCAD (b) Cuttlefish (c) Our method

Fig. 15. Photos of the thin-slab objects (the 5th row of Figure 14) lying on a white table (rows 1–2) and on a black-white edge (rows 3–4).

Figure 16 compares printouts obtained by a remapped 2.5D solution to an inherent 3D solution. For a fair comparison, the native 3D version does not include our improved refinement step (Section 5.5) but employs the same heuristics as the 2.5D method. A comparison with all our features enabled and for all models in Figure 14 is provided in the supplemental.

We observe an overly saturated appearance across the surface of the 2.5D vase (Figure 16a). On thin geometries, such as the hollow neck (marked dashed) or foot (cropout), the surface exhibits unwanted blackening.   The observations are in line with CIEDE2000 average (calculated on renderings) which is 15 for the 2.5D variant and 10 for the 3D one.

Conceptually, the columns in the 2.5D solution get expanded on concave geometry, whereas convex surfaces shrink the columns in depth.   In convex regions, the 2.5D mapping compresses the solution (especially in deeper iso-layers), reducing the fraction of white material in the volume. Since the light absorption is unrecoverable, the resulting appearance is darker and more saturated leading to the false impression of global contrast enhancement. In concave regions, the 2.5D mapping increases the absolute number of absorbing voxels (column expansion) with similar effect. Figure 17 shows how these two effects can even lead to severe color shifts on the cheetah's abdominal.

The third consequence of the remapping is a cutoff in depth due to a smaller thickness of the target geometry. Each side of a wall is by definition only assigned its half of the volume, leading to a cutoff in depth if the overall thickness is below the sum of filled depth on both sides. The planar solution consists of mostly black voxels near the surface, which remain after the remapping, while less absorptive layers underneath are discarded. See examples of this blackening effect in Figure 16c (red arrow) and on the paws in Figure 17. This clipping cannot be solved by simply rescaling the composition in one column to the available depth, as this would induce unwanted brightness and even color shifts.

Our approach (described in Section 5.1) avoids the uncontrolled behavior of the 2.5D method, namely, inaccurate color reproduction and introduced artifacts by operating natively in the final geometric setup and by handling opposing surfaces seamlessly in the refinement step (Section 5.5).

8 CONCLUSION

We presented a technique that handles the inherent limitations of significantly translucent base materials and allows one to compensate lateral scattering in inkjet 3D prints when reproducing textured objects. In contrast to previous approaches, our technique is a genuine 3D approach and is capable of enhancing input textures on curved objects with arbitrary geometry. In the pathological case of very thin geometric features, we propose a preprocessing step that significantly improves consistency and accuracy.

ACKNOWLEDGMENTS

REFERENCES

Vahid Babaei, Kiril Vidimče, Michael Foshey, Alexandre Kaspar, Piotr Didyk, and Wojciech Matusik. 2017. Color Contoning for 3D Printing. *ACM Trans. Graph.* 36, 4, Article 124 (July 2017), 15 pages.

Mary Ann Branch, Thomas F Coleman, and Yuying Li. 1999. A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM Journal on Scientific Computing* 21, 1 (1999), 1–23.

Alan Brunton, Can Ates Arikan, Tejas Madan Tanksale, and Philipp Urban. 2018. 3D Printing Spatially Varying Color and Translucency. *ACM Trans. Graph.* 37, 4, Article 157 (July 2018), 13 pages.

Alan Brunton, Can Ates Arikan, and Philipp Urban. 2015. Pushing the Limits of 3D Color Printing: Error Diffusion with Translucent Materials. *ACM Trans. Graph.* 35, 1, Article 4 (Dec. 2015), 13 pages.

Brent Burley, David Adler, Matt Jen-Yuan Chiang, Hank Driskill, Ralf Habel, Patrick Kelly, Peter Kutz, Yining Karl Li, and Daniel Teece. 2018. The Design and Evolution of Disney's Hyperion Renderer. *ACM Trans. Graph.* 37, 3, Article 33 (July 2018), 22 pages.

Desai Chen, David I. W. Levin, Piotr Didyk, Pitchaya Sitthi-Amorn, and Wojciech Matusik. 2013. Spec2Fab: A Reducer-tuner Model for Translating Specifications to 3D Prints. *ACM Trans. Graph.* 32, 4, Article 135 (July 2013), 10 pages.

Per H. Christensen. 2015. An Approximate Reflectance Profile for Efficient Subsurface Scattering. In *ACM SIGGRAPH 2015 Talks (SIGGRAPH '15)*. ACM, New York, NY, USA, Article 25, 1 pages.

Paolo Cignoni, Enrico Gobbetti, Ruggero Pintus, and Roberto Scopigno. 2008. Color Enhancement for Rapid Prototyping. In *VAST: International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage.*

Eugene D'Eon and Geoffrey Irving. 2011. A Quantized-diffusion Model for Rendering Translucent Materials. *ACM Trans. Graph.* 30, 4, Article 56, 14 pages.

Target   2.5D Printout   3D Printout

(a) Photograph

(b) Target

Printout   Solution

(c) 2.5D Remapped

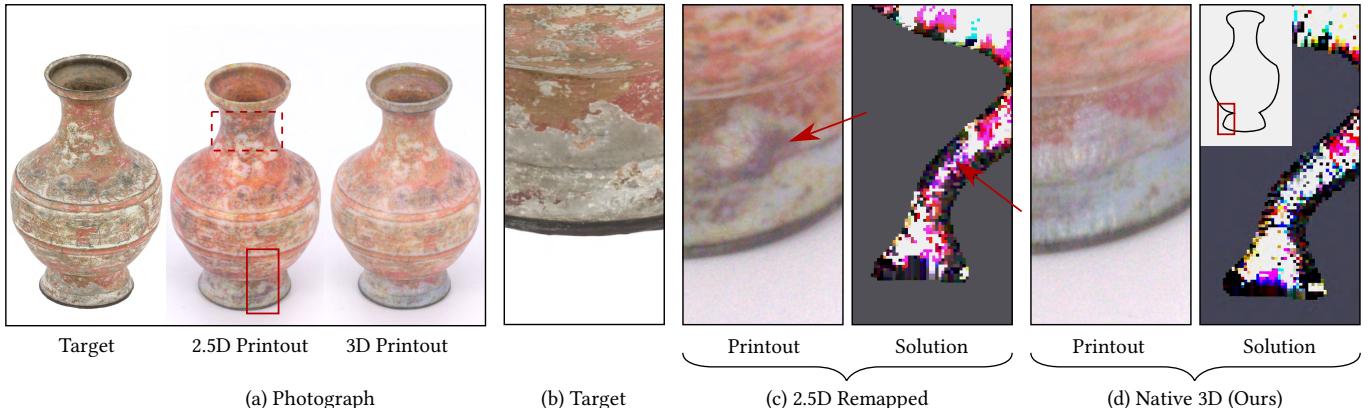Printout   Solution

(d) Native 3D (Ours)

Fig. 16. Baseline Comparison (a) of results achieved with planar scattering compensation (remapped to 3D using UV + depth coordinates) and our native 3D pipeline without our refinement improvements (the new update step and the content-aware gamut mapping). (c) and (d) show cropouts of a region highlighting problems of the 2.5D method and cross-sections thereof in comparison to the target appearance (b).

Fig. 17. Color shifts on the abdomen and blackening on the paws of a 2.5D remapped printout (left) compared to a native 3D printout (right). Similar setup as in Figure 16.

Yue Dong, Jiaping Wang, Fabio Pellacini, Xin Tong, and Baining Guo. 2010. Fabricating Spatially-varying Subsurface Scattering. *ACM Trans. Graph.* 29, 4, Article 62 (July 2010), 10 pages.

Craig Donner, Tim Weyrich, Eugene d'Eon, Ravi Ramamoorthi, and Szymon Rusinkiewicz. 2008. A Layered, Heterogeneous Reflectance Model for Acquiring and Rendering Human Skin. *ACM Trans. Graph.* 27, 5, Article 140, 12 pages.

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 Trans. Graph.* 36, 6, Article 241 (Nov. 2017), 15 pages.

Robert W. Floyd and Louis Steinberg. 1976. An Adaptive Algorithm for Spatial Greyscale. In *Proceedings of Society for Information Displays*, Vol. 17. 75–77.

I. Gkioulekas, A. Levin, and T. Zickler. 2016. An evaluation of computational imaging techniques for heterogeneous inverse scattering. In *European Conference on Computer Vision (ECCV '16)*.

Ioannis Gkioulekas, Shuang Zhao, Kavita Bala, Todd Zickler, and Anat Levin. 2013. Inverse Volume Rendering with Material Dictionaries. *ACM Trans. Graph.* 32, 6, Article 162 (Nov. 2013), 13 pages.

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.

Milovš Hašan and Ravi Ramamoorthi. 2013. Interactive Albedo Editing in Path-traced Volumetric Materials. *ACM Trans. Graph.* 32, 2, Article 11 (April 2013), 11 pages.

Philipp Herholz, Sebastian Koch, Tamy Boubekeur, and Marc Alexa. 2017. Unsharp masking geometry improves 3D prints. *Computers & Graphics* 66 (2017), 135–142.

Roger David Hersch and Frédérique Crété. 2005. Improving the Yule-Nielsen modified Neugebauer model by dot surface coverages depending on the ink superposition conditions. In *Electronic Imaging 2005*. International Society for Optics and Photonics, 434–447.

Wenzel Jakob. 2010. Mitsuba renderer. http://www.mitsuba-renderer.org.

Henrik Wann Jensen, Stephen R Marschner, Marc Levoy, and Pat Hanrahan. 2001. A practical model for subsurface light transport. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. ACM, 511–518.

Yanxiang Lan, Yue Dong, Fabio Pellacini, and Xin Tong. 2013. Bi-scale Appearance Fabrication. *ACM Trans. Graph.* 32, 4, Article 145 (July 2013), 12 pages.

Wojciech Matusik, Boris Ajdin, Jinwei Gu, Jason Lawrence, Hendrik P. A. Lensch, Fabio Pellacini, and Szymon Rusinkiewicz. 2009. Printing Spatially-varying Reflectance. *ACM Trans. Graph.* 28, 5, Article 128 (Dec. 2009), 9 pages.

Mimaki. 2017. Mimaki 3D printer. https://mimaki.com/special/3d_print/. [Online; Accessed 2019-04-25].

Ján Morovic and M Ronnier Luo. 2001. The fundamentals of gamut mapping: A survey. *Journal of Imaging Science and Technology* 45, 3 (2001), 283–290.

Ken Museth. 2013. VDB: High-resolution Sparse Volumes with Dynamic Topology. *ACM Trans. Graph.* 32, 3, Article 27 (July 2013), 22 pages.

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 Trans. Graph.* 32, 4, Article 146 (July 2013), 12 pages.

Ruggero Pintus, Enrico Gobbetti, Paolo Cignoni, and Roberto Scopigno. 2010. Shape enhancement for rapid prototyping. *The Visual Computer* 26, 6-8 (2010), 831–840.

Geoffrey L Rogers. 1997. Optical dot gain in a halftone print. *Journal of Imaging Science and Technology* 41, 6 (1997), 643–656.

Romain Rossier. 2013. Framework for printing with daylight fluorescent inks. (2013).

FR Ruckdeschel and OG Hauser. 1978. Yule-Nielsen effect in printing: a physical analysis. *Applied Optics* 17, 21 (1978), 3376–3383.

SciPy. 2008. scipy.optimize.least_squares SciPy Reference Guide. https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least_squares.html. [Online; Accessed 2019-01-10].

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, Article 271 (Dec. 2018), 15 pages.

Pitchaya Sitthi-Amorn, Javier E. Ramos, Yuwang Wangy, Joyce Kwan, Justin Lan, Wenshou Wang, and Wojciech Matusik. 2015. MultiFab: A Machine Vision Assisted Platform for Multi-material 3D Printing. *ACM Trans. Graph.* 34, 4, Article 129 (July 2015), 11 pages.

Stratasys. 2016. Stratasys J750 the ultimate full-color multi-material 3D printer. https://www.stratasys.com/3d-printers/j735-j750. [Online; Accessed 2019-04-25].

Jiaping Wang, Shuang Zhao, Xin Tong, Stephen Lin, Zhouchen Lin, Yue Dong, Baining Guo, and Heung-Yeung Shum. 2008. Modeling and Rendering of Heterogeneous Translucent Materials Using the Diffusion Equation. *ACM Trans. Graph.* 27, 1, Article 9 (March 2008), 18 pages.

Tim Weyrich, Pieter Peers, Wojciech Matusik, and Szymon Rusinkiewicz. 2009. Fabricating Microgeometry for Custom Surface Reflectance. *ACM Trans. Graph.* 28, 3, Article 32 (July 2009), 6 pages.

Günter Wyszecki and Walter Stanley Stiles. 1982. *Color science: concepts and methods, quantitative data and formulae*. Wiley. 168–169 pages.

Shuang Zhao, Lifan Wu, Frédo Durand, and Ravi Ramamoorthi. 2016. Downsampling Scattering Parameters for Rendering Anisotropic Media. *ACM Trans. Graph.* 35, 6, Article 166 (Nov. 2016), 11 pages.