3D Printing Spatially Varying Color and Translucency

ALAN BRUNTON, CAN ATES ARIKAN, and TEJAS MADAN TANKSALE, Fraunhofer IGD
PHILIPP URBAN, Fraunhofer IGD and Norwegian University of Science and Technology NTNU
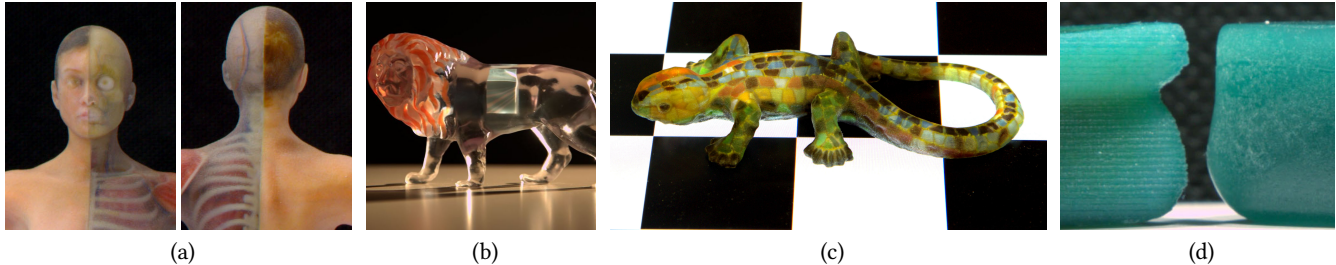
Fig. 1. Design control and perceptual reproduction of material translucency for full color 3D printing: (a) a printed anatomy model with opaque and translucent parts, revealing internal structures; (b) a polished transparent Lion with a red mane and 3D logo inside; (c) a transparent lizard with colored scales illuminated from below by a checkerboard pattern; (d) a close-up (see Figure 12 for the full view) of wax and a 3D print reproducing the wax's translucent appearance.

We present an efficient and scalable pipeline for fabricating full-colored objects with spatially-varying translucency from practical and accessible input data via multi-material 3D printing. Observing that the costs associated with BSSRDF measurement and processing are high, the range of 3D printable BSSRDFs are severely limited, and that the human visual system relies only on simple high-level cues to perceive translucency, we propose a method based on reproducing perceptual translucency cues. The input to our pipeline is an RGBA signal defined on the surface of an object, making our approach accessible and practical for designers. We propose a framework for extending standard color management and profiling to combined color and translucency management using a gamut correspondence strategy we call *opaque relative processing*. We present an efficient streaming method to compute voxel-level material arrangements, achieving both realistic reproduction of measured translucent materials and artistic effects involving multiple fully or partially transparent geometries.

CCS Concepts: • **Computing methodologies → Perception**; *Graphics file formats*;

Additional Key Words and Phrases: 3D printing, metamerism, translucency, color, distance field

Authors' addresses: Alan Brunton, alan.brunton@igd.fraunhofer.de; Can Ates Arikan, arikan.canates@googlemail.com; Tejas Madan Tanksale, tejas.madan.tanksale@igd.fraunhofer.de Fraunhofer IGD; Philipp Urban, philipp.urban@igd.fraunhofer.de Fraunhofer IGD Norwegian University of Science and Technology NTNU.

1 INTRODUCTION

We present an efficient and scalable pipeline for fabricating full-colored objects with spatially-varying translucency from practical and accessible input data via multi-material 3D printing. Color and translucency are encoded by an RGBA signal on the object's surface (i.e. an RGBA texture or per-vertex RGBA attributes), which is supported by various 3D file formats, making our method applicable on many existing models and facilitating model design and creation via existing 3D modeling software.

In contrast to previous approaches focusing only on albedo color fabrication [Babaei et al. 2017; Brunton et al. 2015; Elek et al. 2017; Panozzo et al. 2015; Schüller et al. 2016] or only on translucency fabrication [Dong et al. 2010; Hašan et al. 2010], our method minimizes translucency errors subject to accurate color reproduction. It allows fabricating full-color objects possessing almost opaque, translucent, and fully transparent parts, with both smooth and abrupt changes between these regions. Since many common materials are partially permeable to light such as wax, skin, teeth, cheese, fish, or stone, our method substantially enhances the degree of realism of 3D prints.

Light transport between any two rays that hit the object's surface is phenomenologically modeled by the spatially-varying Bidirectional Surface Scattering Reflectance Distribution Function (BSSRDF) [Nicodemus et al. 1977], intrinsically accounting for spectrally varying absorption and scattering, and phase modulation. A full physical description is thus very high dimension, and multiple works have sought low-dimensional physical approximations [Jensen et al. 2001; Song et al. 2009]. However, by working in the perceptual domain we can significantly reduce the dimensionality even further.

In addition, measuring BSSRDFs [Goesele et al. 2004; Peers et al. 2006; Tong et al. 2005] is expensive in terms of system complexity, acquisition time, storage and processing of captured data. This is true for both natural materials we would like to reproduce, and the output of a fabrication system to characterize the set of BSSRDFs it can produce. There exists no standard format for storing a BSSRDF.

Multi-material 3D printing systems are inherently limited in terms of the set BSSRDFs they can reproduce. This is due to the limited

range of absorption, scattering and refractive indices of available printing materials, the small number of such materials that can be simultaneously combined in a single print, and the limited resolution with which they can arrange these materials.

Thus, both the costs of measuring and processing BSSRDFs, and the physical limitations of printing systems are substantial hurdles to directly using BSSRDFs for graphical 3D printing.

However, human perception of translucency, as revealed by various psychophysical experiments, has little to do with the underlying interactions of light with a given material, and is rather driven by simple high-level cues such as changes in contrast of either the object's surface itself or light from the background passing through the object [Fleming and Bülthoff 2005; Motoyoshi 2010].

In fact, the perceptual notion of translucency encompasses multiple underlying aspects of physical light transport: lateral light transport within the object produces blurring of surface features and albedo color; vertical light transport allows us to see light passing through a highly translucent object from the opposite side.

Our pipeline simultaneously serves two graphical 3D printing applications. First, it allows design control of color and translucency, using existing file formats, which are editable by many existing tools. Second, it allows reproduction of an important part of the perceived translucency of real materials. Our pipeline allows these applications to be unified in the future, by building a library of perceptual translucency values for real materials.

We can take a cue from today's color reproduction systems (printers and displays), which aim to reproduce trichromatic perceptual quantities (lightness, chroma and hue) instead of spectral stimuli. Print reproductions are further restricted to single viewing conditions: one illuminant, a light field, and a viewing geometry. Even though this *metameric* reproduction has systematic limitations, e.g. directional dependencies, color errors are mostly detectible only by a side-by-side comparison with the reference.

We therefore propose to work with *translucency metamers*, reproducing trichromatic (albedo) color attributes and perceptual translucency cues instead of physical quantities of light transport. Toward this end, we make the following technical contributions:

- Combined color and translucency management, including a gamut expansion technique called opaque relative processing, which allows smooth color and translucency look-ups structurally compatible with International Color Consortium (ICC) color management and profiles.
- An efficient streaming algorithm to compute material assignments at the voxel level to fabricate objects with spatially and independently varying albedo color and translucency, which allows color to be preserved as much as possible under varying degrees of translucency.
- An efficient streaming algorithm to compute discrete Voronoi tesselations in $O(1)$ time per voxel.

Streaming algorithms, which we define formally in Section 4, are important in our context for multiple reasons. First, should the printer interface allow it, we can send the relevant control data for the first slices to the printer before completing the full computation. Second, modern voxel-based 3D printing must scale to tens or even hundreds of billions of voxels, given the high resolutions and large build volumes. Streaming algorithms allow us to keep just a small fraction of the total voxels in memory at any time.

2 RELATED WORK

2.1 Color fabrication

There exist commercial 3D printing technologies that allow full-color surface albedo. Powder-binder systems [3DSystems 2014] apply colored adhesives to a nearly opaque white powder substrate, while layer laminate systems [MCor Technologies 2014] function in much the same way, except that the substrate is paper and the inter-layer binding is done using a gluing technique. More recently, a commercial FDM printer has been introduced [XYZ Printing 2017] where the material functions as a substrate for inks which are applied as it is deposited. HP has introduced a powder-based system capable of color [2018], and Morovič *et al.* [2017] proposed a point-process pipeline to combine color with mechanical properties. These systems are limited to color reproduction due to the nearly opaque substrate. More flexible are multi-jet or polyjet systems [Stratasys 2016], in which multiple UV-curing photopolymers are used, and it is possible to create objects with arbitrary material arrangements. This increased flexibility comes at the cost of increase complexity, and the complexity is further exacerbated by the high translucency of the materials compared to the substrate-based technologies.

Recently, data-driven techniques have been proposed, which consider and adapt to the translucency of the printing materials for the purposes of albedo color reproduction. Layered halftoning [Brunton et al. 2015] is an error-diffusion approach for albedo color fabrication, which adapts to the translucency of a given set of printing materials by adjusting the depth underneath the surface to which non-white materials are placed. Further, the scattering behavior of the materials has been considered for improving detail in color texture reproduction on flat surfaces [Elek et al. 2017] using at iterative optimization approach. Multi-material contoning has also been demonstrated using a regression based approach [Babaei et al. 2017]. However, while these works consider the translucency of the available printing materials, they do not provide a mechanism for controlling translucency.

Other fabrication technologies that allow full-color include hydrographics [Panozzo et al. 2015], which allows application of an albedo texture to a 3D object, which may be created by 3D printing, via water transfer printing, and computational thermoforming [Schüller et al. 2016], which uses a similar process, but with a plastic sheet instead of water, to create thin shells of full-color objects. It is not clear how these techniques could be extended to control translucency.

2.2 Fabrication of translucency and scattering

Techniques for both additive and subtractive manufacturing have been proposed [Dong et al. 2010; Hašan et al. 2010] using the scattering profiles approach designed for editing [Song et al. 2009]. These techniqes make use of the Kubelka-Munk layered scattering model [Kubelka and Munk 1931], which considers vertical (forward and back) scattering between layers of different materials (relative to the surface). Lateral light transport can be achieved when the materials have sufficiently different refractive indices, such that a significant portion of light travels laterally within a given layer

(internal reflection). Both works optimize layer thicknesses to approximate a given spatially-varying scattering behavior, although Dong *et al.* consider only flat surfaces, whereas Hašan *et al.* also consider curved surfaces. While both are important works in the field of appearance fabrication, they are not well suited for combining spatially varying color and translucency for the following reasons. First, the Kubelka-Munk model does not describe optically thin materials well, and is therefore not suitable when combining relatively opaque materials with a nearly transparent material, as we do in this paper. Second, Kubelka-Munk assumes infinite lateral dimensions of the layer, which means that color edges or curved geometry break this assumption (the latter effect was observed by Hašan *et al.*). Third, the high-frequency material assignments (halftone) required to achieve full color from a small material set make homogeneous layers impossible.

In contrast, we reproduce degrees of translucency by tuning the mixture of printing materials to adjust the *mean free path length* (MFPL), the expected distance a photon travels within the object before being scattered or absorbed. Details are given in Section 4.2.

Papas *et al.* [2013] use a data-driven approach to optimize pigment concentrations to reproduce homogeneous translucent material appearance, but do not consider spatially-varying translucency and use molds for the fabricated shape. Computational light-routing [Pereira et al. 2014] and printed optics [Willis et al. 2012] exploit differences in refractive indices between printing materials and support material to control light transport within 3D printed objects for user interface and display applications, but do not consider reproduction of physical or perceptual aspects of translucency.

2.3 Perceptual aspects of translucency

Various psychophysical experiments revealed that translucency perception is guided by simple cues rather than the visual system's ability of performing inverse optics on subsurface light transport [Fleming and Bülthoff 2005], i.e. the visual system does not infer the BSSRDF from observed stimuli. Motoyoshi has shown that (particularly high-frequency) luminance contrast and sharpness in non-specular regions of an object serves as a robust cue for our visual system to judge the degree of translucency [Motoyoshi 2010].

Fleming and Bülthoff have shown that "although saturation variations can affect perceived translucency, they are insufficient on their own to yield an impression of translucency" and "...that the saturation component is neither necessary nor sufficient to yield an impression of translucency" [Fleming and Bülthoff 2005]. Fleming and Bülthoff further showed that the effect on luminance contrast of increased translucency changes over the translucency range. For low translucency, contrast decreases as translucency increases, whereas for high translucency, contrast of objects behind the translucent object increases as translucency increases (contrast of the foreground object continues to decrease). This is depicted in terms of the subsurface light transport's *Point Spread Function* (PSF) in Figure 2, and is a perceptual cue we exploit as described in Section 4.2.1.

Gkioulekas *et al.* [2013] have investigated the role of the phase function in the appearance of translucent materials and found that it is significant. They further proposed a 2D space representing many phase functions. Xiao *et al.* [2014] investigated the role of

lighting direction on human perception of translucency and found that humans are able to estimate translucency with consistent accuracy across different shapes and lighting conditions only when the phase function is simple. In particular, isotropic phase functions were perceived with consistent translucency under different lighting and geometry. Recently, Urban *et al.* [2017] proposed a 1D family of materials with isotropic phase functions, and used a psychometric function to embed them in a perceptually uniform scale. This family of materials can be used as a subtractive mixing interpretation of the alpha channel of RGBA textures, as we do in this paper.

2.4 Fabrication pipelines

Programmable [Vidimče et al. 2013] and specification-driven [Chen et al. 2013] fabrication pipelines for multi-material 3D printers have been proposed to combine clear and opaque materials, and to implement Kubelka-Munk scattering fabrication [Hašan et al. 2010].

2.5 Distance field computation

In Section 4.1 we describe an efficient streaming approach to discrete distance field computation and Voronoi tesselation of the voxel grid. Distance field computation in general is a well-studied problem and we refer the reader to Jones *et al.* [2006] for a survey. Our technique has similarities to fast sweeping methods [Detrixhe et al. 2013; Zhao 2004, 2007], and uses distance transforms [Felzenzwalb and Huttenlocher 2012]. Fast sweeping, like fast marching [Sethian 1999], allows non-constant metrics, but is linear in the number of voxels. Neither fast sweeping nor fast marching nor direct use of distance transforms are compatible with streaming; they require the full voxel grid to be stored at once. Our approach is limited to Euclidean metrics, but an efficient pre-process allows us to compute distance fields a slice at a time. Our technique further also allows to compute overlapping distance fields, as described in Section 4.1.

The streaming requirement is less well studied. An exception is Sud *et al.* [2006], who use linear factorization to accelerate distance field computation on the GPU, with a technique that allows per-slice computation. This technique relies strongly on GPU hardware accelerated operations, and its complexity grows with the geometric complexity of the input (number of primitives). The algorithm presented in Section 4.1 has computational complexity independent of the geometric complexity of the input and grows linearly with respect to the number of voxels. Like Sud *et al.*, recent work has focused on GPU acceleration. For example, Liu and Kim [2014] present GPU-accelerated bounding volume hierarchy (BVH) construction and sampling, though they evaluate distances on adaptive grids with sparse samples. Vidimče *et al.* [2013] use a BVH (octree) in their pipeline and note the distance computation to be a bottleneck, though they did not exploit GPU acceleration. While we have implemented our algorithm using CPU multi-threading, each step is parallel and could be further accelerated by a GPU implementation.

3 BACKGROUND AND OVERVIEW

3.1 Physical Light Transport and Human Perception

The BSSRDF $\mathbf{B}_\lambda \in \mathbb{B}$ describes the phenomenon of light transport in a non-fluorescent and non-self-luminous object [Nicodemus et al.

1977], which allows us to compute the radiance $\mathbf{L}_\lambda$ emitted from the object given the irradiance $\mathbf{E}_\lambda$ at each point on the surface.

The radiance emitted from the object and observed by a subject is called stimulus. The transformation $\Theta_{E,X} : \mathbb{B} \mapsto \mathbb{S}$ takes a BSSRDF $\mathbf{B}$ and produces a stimulus $\mathbf{S}$ given the spatially-varying irradiance E and the viewing position X of the subject. See the supplemental appendix for details.

The stimulus $\mathbf{S} = \Theta_{E,X}(\mathbf{B})$ is then processed by our visual system, which spatially and spectrally filters, attenuates and amplifies the signals yielding the perceived visual appearance of the object and its material. Though the human visual system is far from being understood, various psychophysical models were proposed to describe the visual pathway and processing of the signals as well as the representation of perceived appearance [Palmer 1999]. Such a psychophysical model $\Gamma_\mathbb{A} : \mathbb{S} \mapsto \mathbb{A}$ transforms the stimulus $\mathbf{S}$ into perceptual correlates $\mathbf{A}$ describing our perception of the object's appearance (*e.g.* color, gloss, translucency) [Hunter and Harold 1987].

In summary, the perceived appearance of an object BSSRDF $\mathbf{B} \in \mathbb{B}$ can be described by a function composition $\Gamma_\mathbb{A}(\Theta_{E,X}(\mathbf{B}))$ considering the irradiance E and the position X of the subject.

3.2 Translucency Metamers

Our approach is based on the observation that in many applications accurately reproducing some perceptual correlates is more important than others, particularly if the output device is limited in reproducing some of the correlates. One example is color: from five color attributes, today's printers and displays reproduce only *lightness*, *chroma* and *hue* [Fairchild 2005]. The remaining attributes, *brightness* and *colorfulness*, depend on the absolute luminance and most devices are very limited in reproducing low or high luminance levels. Therefore, these color attributes are almost never considered. Lightness, chroma and hue are sufficient for most applications.

We make the assumption that for graphical 3D printing, it is more important to accurately reproduce color than translucency.[1] To our knowledge, psychophysical experiments to confirm or refute this assumption have never been conducted, though we believe it is a reasonable one. To expand 3D color printing by spatially-varying translucency, we propose the concept of *translucency metamers*, which are materials with potentially different translucency attributes but similar color attributes for particular viewing conditions. By adjusting the color of the 3D print and using the remaining degrees-of-freedom of material placement to minimize perceived translucency differences between input and print, our approach optimizes the reproduction of perceptual attributes linked to translucency metamers instead of physical quantities.

Nevertheless, our method can also be described by BSSRDFs and extended beyond the reproduction of color and translucency. For this, we define an equivalence relation between object BSSRDFs,

$$\mathbf{B}_1 \underset{\mathbb{A}, E, X}{\sim} \mathbf{B}_2 :\Leftrightarrow \Gamma_\mathbb{A}(\Theta_{E,X}(\mathbf{B}_1)) = \Gamma_\mathbb{A}(\Theta_{E,X}(\mathbf{B}_2)) \quad (1)$$

which means the two object BSSRDFs are equivalent if considered perceptual correlates match under the specified viewing conditions E, X for the psychophysical model $\Gamma_\mathbb{A}$. This defines an equivalence class of object BSSRDFs $[\mathbf{B}]_{\mathbb{A}, E, X} := \{\mathbf{C} \in \mathbb{B} \mid \mathbf{B} \underset{\mathbb{A}, E, X}{\sim} \mathbf{C}\}$. We refer

---

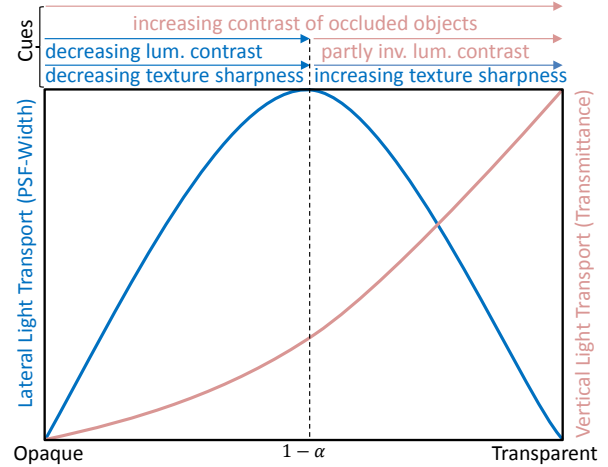[1]The opposite may well be true for printing optics.

Fig. 2. Schematic relationship between lateral and vertical light transport and its impact on translucency cues for the case that only scattering increases with α.

to these as BSSRDF *metamers* for the given viewing conditions. This definition is an expansion of reflectance metamers used in 2D printing. In turn, each $\mathbf{A} \in \mathbb{A}$ defines a set of BSSRDF metamers by $[\mathbf{A}]_{E,X} := \{\mathbf{C} \in \mathbb{B} \mid \mathbf{A} = \Gamma_\mathbb{A}(\Theta_{E,X}(\mathbf{C}))\}$. Note that $[\mathbf{B}]_{\mathbb{A}, E, X} = [\Gamma_\mathbb{A}(\Theta_{E,X}(\mathbf{B}))]_{E,X}$.

The concept of metamerism allows us to state the relationship between the BSSRDF $\mathbf{B}_{\text{print}}$ of the print created by our approach and the BSSRDF $\mathbf{B}$ yielding the RGBA input for viewing conditions E, X:

$$\mathbf{B}_{\text{print}} = \underset{\mathbf{C} \in [\mathbf{F}(\mathbf{A_B})]_{E,X}}{\operatorname{argmin}} \mathbf{d}(\Gamma_\mathbb{T}(\Theta_{E,X}(\mathbf{C})), \Gamma_\mathbb{T}(\Theta_{E,X}(\mathbf{B}))) \quad (2)$$

where $\mathbb{T}$ is the translucency attribute space, $\mathbf{d}$ is a metric reflecting the perceived translucency difference, $\mathbf{A_B} = \Gamma_\mathbb{C}(\Theta_{E,X}(\mathbf{B}))$ with $\mathbb{C}$ being the color attribute space for lightness, chroma and hue, and $\mathbf{F} : \mathbb{C} \mapsto \Gamma_\mathbb{C}(\Theta_{E,X}(\mathbb{G}))$ is a color gamut mapping transformation with $\mathbb{G}$ being the set of BSSRDFs printable by the device. $\mathbf{F}$ is required because printing systems are inherently limited in reproducing color attributes even for fixed viewing conditions.

In this work, we consider off-specular 0/45 illuminating/viewing conditions, the **CIELAB** color space for $\mathbb{C}$ (lightness, chroma and hue can be predicted by a cylindrical coordinate transformation), and a perceptually-uniform α-scale for the translucency space $\mathbb{T}$ [Urban et al. 2017], i.e. we use the Euclidean metric for $\mathbf{d}$ in (2).

3.3 Workflow

Fig. 3 shows a high-level view of our workflow to reproduce color and translucency by 3D printing via translucency metamers. We now briefly discuss the main components.

(1) Shape + RGBA: The input to our pipeline is one or more shapes $\mathcal{S}_i \subset \mathbb{R}^3$, $i = 1, \ldots, n$ defined by a surface $\partial \mathcal{S}_i$ (*e.g.* a mesh) with attached RGBA signal[2]. We interpret the α-channel as defined by Urban *et al.* [2017].

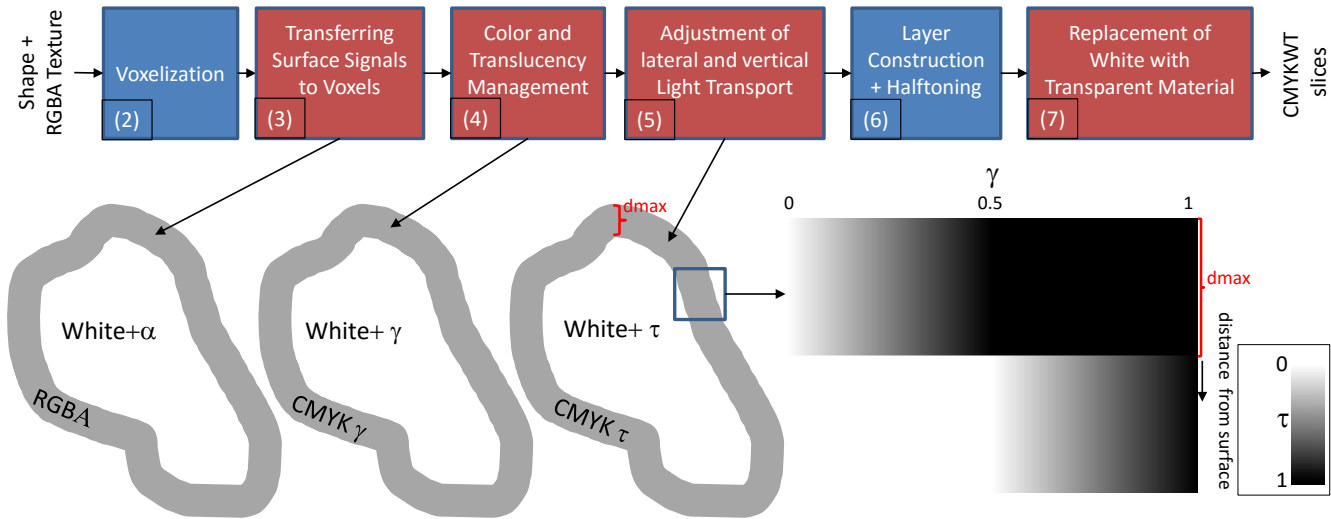[2]we interpret RGB as sRGB and A stands for the translucency-channel denoted as α throughout the paper.

Fig. 3. Our workflow for combined color and translucency reproduction. Red boxes indicates parts of the pipeline containing our contributions.

(2) Voxelization: We convert the mesh into voxel data distinguishing surface, interior and exterior voxels. Surface voxels are stored in a sparse representation with the associated input RGBA values.

(3) Transferring Surface Signals to Voxels: In this phase, we compute the distances of inner voxels to the surface and assign RGBA values to the full interior of the object. (Section 4.1.)

(4) Color and Translucency Management: We employ gamut mapping and a multidimensional lookup table computed offline (Section 5.4) to map RGBA values to printer control (tonal) values.

(5) Adjustment of Lateral and Vertical Light Transport: To increase the gamut of perceptual translucency cues we split the range of translucency values into two: starting from maximally opaque, the light transport's PSF widens until a threshold, after which it narrows again towards maximally transparent, as in Fig. 2. (Section 4.2.1.)

(6) Layer Construction and Halftoning: For color halftoning we use the method of Brunton *et al.* [2015]. Note that translucency values are not halftoned, only color tonal values.

(7) Replacement of White with Transparent Material: Following halftoning, voxels assigned white material are replaced probabilistically with transparent material. (Section 4.2.2.)

(8) Material Arrangement: The output of our pipeline is a voxel-level material arrangement $m : \mathcal{V} \mapsto \mathcal{M}$, where $\mathcal{V}$ is the set of voxels belonging to the input shape(s), the set of materials $\mathcal{M}$ is discrete and small−6 in the case of CMYKWT materials. We compute m per-slice, and send it to the device.

4 STREAMING VOXEL-BASED PIPELINE

We generate printer control data for objects with translucency metamers using a streaming voxel-based pipeline. We now describe the key aspects of this pipeline as they relate to our core contributions.

We define a streaming computation in the context of voxel-based additive manufacturing as follows.

Definition 4.1. Since the manufacturing process builds the object from bottom to top, we define streaming computation to proceed in ascending order of z. We further define that for a given print occupying a volume corresponding to a voxel grid of $W \times H \times D$ voxels, a streaming algorithm never exceeds $O(WH + N)$ storage, where $N \ll WHD$; *i.e.* a constant number of slices plus small auxiliary storage, relative to the size of the voxel grid.

We use the term *chunk* to refer to a constant number of slices that are generated or processed at a time. The number of slices in a chunk is determined by the amount of buffering required for different computations. For our pipeline this works out to 22 slices.

A natural use of translucency is to overlap or nest objects, with outer objects being made varying degrees of translucent, revealing the internal structures. Examples of this include Fig. 9 and 10. To allow this, we assign each object a unique ID and voxels are assigned the ID of the object to which they "belong" (an ID of 0 indicates an empty voxel). We resolve conflicts for voxels inside multiple objects by assigning each object a priority and processing them in order of priority, assigning voxels on a first-come, first-served basis. This priority can be controlled by the user, or the object with the smallest bounding box can be assigned the highest priority.

4.1 Volumetric Color and Translucency

Since translucency is a long-range effect, especially when tending towards transparency, this information needs to be available throughout the volume of the object, *i.e.* at every voxel. However, our input is only the surface with attached color and translucency, typically a texture. Hence we need to transfer this information from the surface to every point inside the object.

It is not well defined how translucency information on the surface should relate to translucency information throughout the volume, and we therefore introduce the following heuristic. We assign to every interior voxel the alpha value of the nearest point on the surface. Though simple, this avoids additional assumptions about the relationship between interior and surface properties, which is anyways built into the profile during printer characterization, as described in Section 5. It further ensures that the volumetric translucency signal deviates significantly from that of the surface only at the object's medial axis. We use the same process for color: we transfer the full RGBA value to a surface point's entire Voronoi cell. We compute this efficiently in terms of time and space as follows.

We first perform a streaming 3D rasterization of each surface ∂S_i, using an algorithm similar to that of Schwarz and Seidel [2010] for 6-separating surface voxelizations. We use a 6-separating surface voxelization to increase sparsity of our surface data. We rasterize the surface a chunk at a time. The rasterization process interpolates per-vertex RGBA values or texture coordinates at each voxel intersected by the surface, batches these voxels together and applies a GLSL shader to sample the texture or generate procedural RGBA values. We allow only one object to rasterize to a voxel, resolving conflicts via object priority. After each chunk, the rasterized surface voxels are stored in a set of ordered lists, one per (x, y) position in the voxel grid per object. Each entry stores the z value and the RGBA value; we denote the lists by $\mathbf{Z}_i(x, y)$ and $\mathbf{RGBA}_i(x, y)$. Empty lists are pruned when the rasterization has completed for all slices. A similar data structure, storing only the z values, has been used to optimize layer thickness [Alexa et al. 2017].

The above data structure admits an efficient, $O(1)$ per voxel, streaming algorithm to compute, for all voxels, the distance to the nearest surface voxel, and to transfer the RGBA value to its entire discrete Voronoi cell. Because we use the distance to surface to control material assignments, in the case of overlapping objects we must compute the distance to the nearest surface voxel, which belongs to the same object (same ID).

The algorithm proceeds in three steps for a given slice s, which corresponds to z-coordinate z_s in the print volume. Let $o_s(x, y)$ denote the object ID of the given voxel in slice s as computed during voxelization and id_i the ID of object S_i.

(1) Initialize the distance and RGBA values for the slice:
$d_s(x, y) = \infty$, $RGBA_s(x, y) = (1, 1, 1, 1)$, for all (x, y).

(2) For $i = 1, \ldots, n$
   (a) for all (x, y) s.t. $\mathbf{Z}_i(x, y) \neq \emptyset$,
       find $z \in \mathbf{Z}_i(x, y)$ s.t. $(z - z_s)^2$ is minimized, and let $k(x, y)$ be the corresponding index.
   (b) if $o_s(x, y) = id_i$ then assign
       $d_s(x, y) = (z - z_s)^2$ and $RGBA_s(x, y) = \mathbf{RGBA}_i(x, y)[k(x, y)]$.

(3) Apply a *multi-track* 2D squared Euclidean distance transform (DT) to $d_s(x, y)$, using a modified version of the algorithm of Felzenzwalb and Huttenlocher [2012], such that when the $d_s(x, y)$ is overwritten, so is $RGBA_s(x, y)$. By multi-track, we mean that 1D DTs along x and y (lower envelopes of parabolas) are computed separately for each id_i, or track, and $o_s(x, y)$ is used to look-up the correct track. Effectively, n independent 1D DTs are computed along each row and column of the slice. The algebraic

representation of the lower envelope in the 1D DT allows to skip voxels belonging to a different object. Please see Felzenzwalb and Huttenlocher for details of the lower envelope construction.

CLAIM 1. *Following step (3), $d_s(x, y)$ contains the squared Euclidean distance to the nearest surface voxel of ∂S_i s.t. $id_i = o_s(x, y)$ for all voxels in slice s. Further, $RGBA_s(x, y)$ contains the RGBA value of that nearest surface voxel.*

CLAIM 2. *When processing slices in ascending order of z_s, the above algorithm runs in $O(1)$ time per voxel, and satisfies the storage requirements of Definition 4.1.*

Though straightforward, for completeness we provide the proofs in the supplemental appendix.

While a 3D distance transform would achieve the same run-time, it would require storing the full voxel grid simultaneously. Other pre-computed structures, *i.e.* based on space partitioning would allow computing distance for only a subset of voxels in a slice, but do not achieve constant query time, depending rather on the input.

We now have RGBA values throughout the volume of the object. Next, we convert these into printer control, or tonal, values. This is done efficiently via a multi-dimensional lookup table, which is precomputed by the joint color and translucency management described in Section 5. We convert printer control values into discrete material assignments as described in Section 4.2.

4.2 Material Arrangement

We focus on multimaterial 3D printers utilizing cyan (C), magenta (M), yellow (Y), black (K), white (W) and a transparent (T) material. Our approach needs a white and a transparent material with negligible absorption, but it is not restricted to CMYKWT; more color materials can be employed to enlarge the color gamut. Black helps increase the dynamic range of the print, but is not required for full-color printing.

For color printing, we use layered halftoning [Brunton et al. 2015], which has been shown to produce high quality color material assignments for multi-material 3D printers. Other material arrangement techniques, such as contoning, could also be used.

Layered halftoning assigns colored materials only up to a depth $0 < d_{near} \leq d_b$ beneath the surface, where d_b is a sufficient depth to achieve the darkest possible color with the given set of materials. Using d_{near} instead of d_b provides an empirical trade-off between computational effort and minimized reflectance. We exploit this definition of a near-surface region in our translucency control.

To create a distinct degree of translucency, we replace only white material with transparent. The arrangement of CMYK materials stays unchanged for given CMYK tonal values. Since white and transparent materials both have negligible absorption coefficients over the visible spectrum, light passing through them changes its spectral power distribution (SPD) negligibly. Transparent material also has a negligible scattering coefficient causing light passing through to keep its direction. White material has a high scattering coefficient and a much greater fraction of incident light is scattered into a different direction. This concept is depicted in Fig. 4. Replacing white with transparent material locally increases the MFPL.
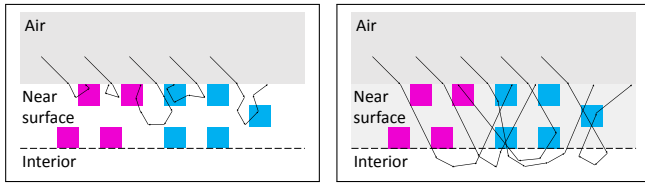
Fig. 4. Conceptual diagram showing increased MFPL by replacing white with clear in the near surface region. Left: The near surface region is filled with white material for opaque color printing. Light travels short distances before being scattered and tends to exit the surface close to where it entered. Right: White is replaced with clear, which reduces scattering. Light travels farther before being scattered, and the SPD is modulated by more color materials, resulting in a blurred appearance compared to the opaque case.

4.2.1 Lateral vs. Vertical Light Transport. The printer is controlled by **CMYK**γ tonal values, where $\gamma \in [0, 1]$ is a fraction of white material to be replaced by transparent material. Setting $\gamma = 0$ achieves the maximum opacity print, equivalent to a standard color print. For $\gamma = 1$ all white material is replaced by transparent.

Rather than replacing white with clear uniformly throughout the object, for small values of γ, we replace white voxels in the near surface region and replace voxels deeper inside the object only for large values of γ. Specifically, we create a mapping $\gamma \times d \mapsto \tau$, which maps γ values at different distances to the surface to values τ, which correspond to the probability of replacing white material with transparent material. Formally,

$$\tau(\gamma, d) = \begin{cases} \min(t, \gamma)/t & \text{if } d < d_{\text{near}} \\ \max(0, \gamma - t)/(1 - t) & \text{otherwise} \end{cases} \quad (3)$$

where t is a threshold separating γ into lower and higher ranges. We use $t = 0.5$ in this paper. Thus, we replace all white in the near-surface region with transparent before beginning to replace in the interior of the object, as shown in the bottom right of Fig. 3.

This choice is motivated by several factors. First, by replacing white voxels in the near surface region before those in the deep interior, we can increase the gamut of translucency cues. The deeper inside the object, the more light has already been attenuated, especially for small γ. Locally increasing the MFPL deep inside the object will have negligible effect on its perceived translucency. Since d_{near} is set near the limit of visible attenuation differences, we can expect increasing the MFPL in this range to have a visible impact.

Second, we want to replicate as closely as possible the perceptual cue depicted in Fig. 2: changes in contrast behavior over the translucency range. For $\gamma \le t$, we main a white backing, while allowing light to travel further in the near-surface region before being scattered back to the surface. This increases the probability that the SPD of the light is modulated by traveling though different non-white materials, resulting in more uniform absorption, and thus reduced contrast and blurring of details. This effect is shown conceptually in Fig. 4, and in practice for real prints in Fig. 8 for color checkers with different α values and in Fig. 11, where we can see how geometric and texture detail is blurred for smaller α.

For $\gamma > t$, we begin to replace white beyond the near-surface region, allowing light to travel deeper into the object, and eventually

through it for $\gamma = 1$. This increases the likelihood that light from a nested object behind the surface passes through without being modulated, increasing its observed contrast.

Third, maintaining a white backing helps better preserve color for small γ. Table 2 shows color error statistics for different α.

4.2.2 Probabilistic transparent material assignment. We replace white with transparent material using a dart-throwing approach w.r.t. τ. For each voxel assigned white material by the halftoning algorithm, we generate a random number u uniformly distributed over $[0, 1)$. If and only if $u < \tau$, we switch the material to clear.

Even though absorption of white and transparent material is almost zero, the relative SPD of emitted light from the same CMYK material arrangement changes with γ. This is because a fraction of light incident at any location of the object contributes to the emitted light after its SPD is modulated by transited CMYK materials. Changing γ changes the MFPL and thus the mixture of light. In general, this causes chroma and hue shifts. Preserving the albedo color at different translucency levels requires an adjustment of the CMYK tonal values, which is done in the lookup table (Section 5.4).

5 COLOR AND TRANSLUCENCY MANAGEMENT

5.1 Measurements

Color is measured on a flat surface in an off-specular 0/45 geometry ensuring that the detection area is much smaller than the illuminating area to avoid *edge loss* – the drop of radiance emitted from the surface at the edge of the detection area due to subsurface light transport. We follow the approach proposed by Arikan *et al.* [2015] and use an almost colorimetric DSLR camera [Canon 2014] and a spectrally tunable LED light source [Image Engineering 2015] whose SPD is adjusted to mimic CIED50 illuminant. Captured RGB data was corrected for dark current noise, flat fielded and mapped to CIEXYZ values using a second-degree polynomial approach.

For translucency, we used the definition of α proposed by Urban *et al.* [2017]. For more details, please refer to that paper, but we provide a brief overview for completeness. This definition relies on a set of virtual reference materials with an isotropic phase function and varying absorption and scattering coefficients. Reference materials are assigned to α values aiming to embed them into a nearly perceptually-uniform 1D translucency space. The α of a real material is determined by finding the reference material for which measurements of lateral and vertical light transport are most similar.

We measure lateral and vertical light transport for specified measurement conditions using a transmission / reflection spectrophotometer [Barbieri 2015]. Vertical light transport is measured in transmission mode for which the 4 mm-thick sample is placed between light source and detector. Lateral light transport is measured in reflectance mode (detector and light source are on the same side of the sample in a 45/0 geometry) employing the spectral edge-loss difference [Yoshida et al. 2011], which requires two spectral measurements both with the same detection area (circular with 2mm diameter) but with two different illuminating areas. This is achieved by using two circular apertures: one with 2mm and one with 8mm diameter. For light-scattering materials, edge loss of the $2mm/2mm$ setup is always larger than the $2mm/8mm$ setup and the difference of edge loss is a simple measure of lateral light transport.

5.2 Optical Printer Model

The optical printer model is a predicting function of the printer's color and translucency output for given tonal values, i.e. $\mathbf{f} : \mathbf{CMYK}\gamma \mapsto \mathbf{CIELAB}\alpha$. In this work, we use a broadband cellular Neugebauer model [Rolleston and Balasubramanian 1993] in intensity linear $\mathbf{CIEXYZ}\alpha$ space, i.e. we also predict α by this model. $\mathbf{CIEXYZ}\alpha$ are then further transformed to $\mathbf{CIELAB}\alpha$. Linearization of the CMYK tonal values is performed using the broadband Murray-Davies model according to [Wyble and Berns 2000] at each γ level. This means that in contrast to color-only printing that uses one-dimensional LUTs to convert nominal to effective tonal values, we use two-dimensional LUTs considering also the actual γ-level.

In this work, we employed $4^5 = 1024$ cells requiring a total of $5^5 = 3125$ printed training patches for fitting the cellular Neugebaur model. The targets contained all elements of $\{0, 0.25, 0.5, 0.75, 1\}^5 \subset \mathbf{CMYK}\gamma$. We used $4 \times 11^2 = 484$ patches for linearization, i.e. for each $H \in \{C, M, Y, K\}$, we printed a target with patches $(H, \gamma) \in \{0, 1/10, \ldots, 1\}^2$ keeping the other color tonals zero. We used linear interpolation to create the four two-dimensional linearization LUTs.

5.3 Media and Opaque Relative Processing

Media relative processing ensures that an input white, RGB = (1,1,1), is mapped to the white printing material, CMYK = (0,0,0,0), even if the color of the white material is not the perfect reflecting diffuser, i.e. its color does not match CIELAB = (100,0,0). This approach assumes that the human visual system adapts to the illuminant emitted from the white printing material instead of the color of the illuminating illuminant. Media relative processing can be interpreted as a first global gamut mapping in color management accounting for reflectance limitations of real white printing materials – their absorption is very small but not zero. It is performed by a simple linear scaling of the white material's color to the illuminant's color in the intensity-linear CIEXYZ color space, i.e. the media relative color gamut includes CIELAB = (100,0,0), a property that is considered by subsequent gamut mapping methods. For characterizing the printer, we perform this media relative scaling to all measured color data.

For translucency, we follow the same idea by clipping *absolute* α-values to $[\alpha_1, \alpha_0]$ and scaling them by $\alpha' = (\alpha - \alpha_1)/(\alpha_0 - \alpha_1)$, where α_1 and α_0 are the white point's α-values for $\gamma = 1$ and $\gamma = 0$. As described in 4.2, lightness drops with increasing γ, i.e. it is impossible to preserve it for translucent/transparent prints. Therefore, we adjust the α'-scale to satisfy a linear relationship with the white points' media relative lightness values across $\gamma \in [0, 1]$. This ensures that a linear gradient of the resulting *relative* α-values for whites (CMYKγ = (0,0,0,0,γ), $\gamma \in [0, 1]$) corresponds to a linear lightness gradient.

Since preserving relative lightness contrast is more important than preserving absolute lightness [Lissner et al. 2013; Wang et al. 2004], we propose *opaque relative processing* for the color dimensions that reduces lightness deviations between different translucency levels. We achieve this by subsequent lightness scaling and gamut expansion of $\mathbb{G}_\gamma \subset \mathbf{CIELAB}$ towards $\mathbb{G}_0 \subset \mathbf{CIELAB}$, where $\mathbb{G}_\gamma$ is the media relative color gamut belonging to γ.

(1) We scale lightness values for each color in $\mathbb{G}_\gamma$, $\gamma > 0$, as follows

$$L^*_{\gamma, \text{or}} = \max \left\{ L^*_{\gamma, \text{mr}}, \frac{100}{a-1} \left(a^{\frac{L^*_{\gamma, \text{mr}}}{L^*_{\gamma, \text{mrw}}}} - 1 \right) \right\} \quad (4)$$

where $L^*_{\gamma, \text{mr}}$ is the media relative lightness, $L^*_{\gamma, \text{mrw}}$ is the media relative white point's lightness and $a = (1/Y_{\gamma, \text{mrw}})^3$ with the media relative white point's luminance $Y_{\gamma, \text{mrw}}$ (corresponding to $L^*_{\gamma, \text{mrw}}$). Fig. 5 illustrates the scaling for different γ values. To ensure that the opaque relative white point $(L^*_{\gamma, \text{orw}}, a^*_{\gamma, \text{orw}}, b^*_{\gamma, \text{orw}})$ corresponding to $(0, 0, 0, 0, \gamma) \in \mathbf{CMYK}\gamma$ is mapped to CIELAB $= (100,0,0)$, we alter the media relative chromatic components $(a^*_{\gamma, \text{mr}}, b^*_{\gamma, \text{mr}})$ as follows

$$\begin{pmatrix} a^*_{\gamma, \text{or}} \\ b^*_{\gamma, \text{or}} \end{pmatrix} = \begin{pmatrix} a^*_{\gamma, \text{mr}} \\ b^*_{\gamma, \text{mr}} \end{pmatrix} - \frac{L^*_{\gamma, \text{or}} - L^*_{\gamma, \text{mr}}}{100 - L^*_{\gamma, \text{mr}}} \begin{pmatrix} a^*_{\gamma, \text{mrw}} \\ b^*_{\gamma, \text{mrw}} \end{pmatrix} \quad (5)$$

where $(a^*_{\gamma, \text{mrw}}, b^*_{\gamma, \text{mrw}})$ are the media relative chromatic components corresponding to $(0, 0, 0, 0, \gamma) \in \mathbf{CMYK}\gamma$. We denote the gamut of all colors in $\mathbb{G}_\gamma$ lightness-scaled according to eq. (4) and (5) by $\mathbb{G}'_\gamma$.

(2) The gamut expansion $\mathbf{e}_\gamma : \mathbb{G}'_\gamma \mapsto \mathbf{CIELAB}$ is performed by a hue-preserving, star-shaped linear expansion from the middle gray value $\mathbf{m} = (50, 0, 0) \in \mathbf{CIELAB}$ in a hue-linearized **CIELAB** color space (see Fig. 6). Each color $\mathbf{x} \in \mathbb{G}'_\gamma$ is mapped along the line $\mathbf{l}(\lambda) = \mathbf{x} + (1 - \lambda)(\mathbf{m} - \mathbf{x})$, $\lambda \geq 0$ as follows

$$\mathbf{e}_\gamma(\mathbf{x}) := \begin{cases} \mathbf{l}(\lambda_0/\lambda_\gamma), & \lambda_\gamma < \lambda_0 \\ \mathbf{x}, & \text{else} \end{cases} \quad (6)$$

where $\mathbf{l}(\lambda_0) \in \partial\mathbb{G}_0$ and $\mathbf{l}(\lambda_\gamma) \in \partial\mathbb{G}'_\gamma$ are the intersections of the line with the gamut boundaries.

We denote the opaque relative transformation by $\mathbf{q} : \mathbf{CIELAB}\alpha \mapsto \mathbf{CIELAB}\alpha$. The opaque relative representation ensures that each color in the printer's opaque gamut $\mathbb{G}_0$ can be reproduced by $\mathbf{CMYK}\gamma$ values containing every γ-level. This allows us to create smooth lookup tables from $\mathbf{CIELAB}\alpha$ to $\mathbf{CMYK}\gamma$ (Sec. 5.4) required to avoid color and translucency banding artifacts. Note that hues are almost preserved in all processing steps, up to slight hue changes due to white point adjustment in (5), which is an important heuristic used in gamut mapping algorithms to avoid memory color mismatches.

5.4 Color and Translucency Lookup Table

This lookup table encodes the transformation $\mathbf{p} : \mathbf{CIELAB}\alpha \mapsto \mathbf{CMYK}\gamma$. It is a concatenation of a *gamut mapping* transformation that maps all $\mathbf{CIELAB}\alpha$ values into the color and translucency gamut of the device and a *separation* transformation that transforms each in-gamut $\mathbf{CIELAB}\alpha$ value to a $\mathbf{CMYK}\gamma$ print control value that reproduces it. The latter is in general ambiguous since multiple $\mathbf{CMYK}\gamma$ values can produce the same $\mathbf{CIELAB}\alpha$ output.

Computing the table requires the optical printer model $\mathbf{f}$ from Sec. 5.2. The predicted $\mathbf{CIELAB}\alpha$ values are then further processed by the opaque relative transformation $\mathbf{q}$ from Sec. 5.3. In the following the function $\mathbf{h} := \mathbf{q} \circ \mathbf{f} : \mathbf{CMYK}\gamma \mapsto \mathbf{CIELAB}\alpha$ is used to characterize the printing system.

We encode $\mathbf{p}$ by a lookup table compatible with the ICC standard allowing us to use a standard color management module (CMM) to
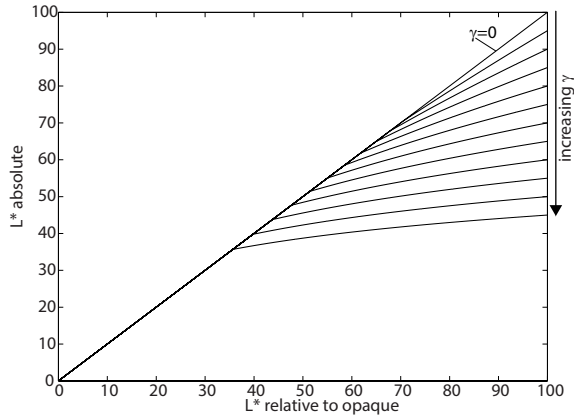
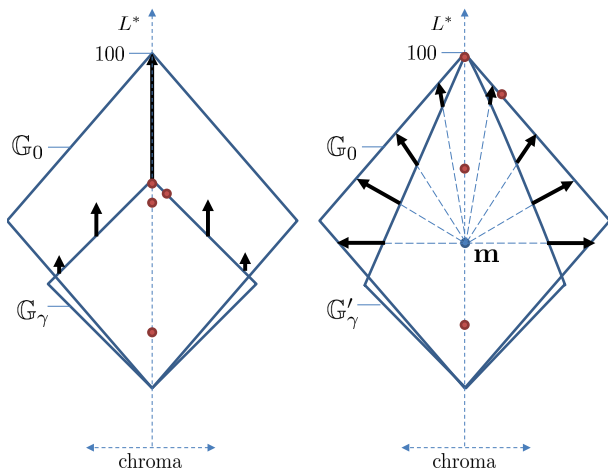Fig. 5. Compressing lightness in Opaque Relative Workflow

Fig. 6. Opaque relative processing: Lightness scaling (left) and gamut expansion (right). Red dots show corresponding colors for the same γ-level in media relative (left) and opaque relative representation (right).

compute **CMYK**γ values from input **CIELAB**α. We give the procedure we use to compute the table in the supplemental appendix.

Note that the lookup table encodes the media relative color and translucency transformation. For reproducing absolute quantities, media relative processing transformations must be applied to the input color and translucency before the table lookup (similar to the *absolute rendering intent* processing in ICC color measurement).

6 EXPERIMENTS

We validate our approach along three axes. First, by showing the stability of in-gamut color accuracy under translucency change and the accuracy of in-gamut translucency reproduction. Second, by showing its ability to generate 3D prints with spatial control over translucency, including hue-preserving smooth gradients and sharp edges. Third, we show its ability to reproduce the perceived translucency of measured materials. We show additional examples in the

supplemental appendix and provide the input for our experiments in the supplemental material.

We carry out our evaluation with the following hardware setup: as a 3D printing device we use a Stratasys J750 [2016] with materials VeroCyan, VeroMagenta, VeroYellow, VeroBlack, VeroPureWhite and VeroClear. We use the printer's 27 μm mode (slice thickness) and generate slices at the native resolution of 42 μm in x and 84 μm in y, and drive the printer at the voxel level [Stratasys 2017]. All computations were performed on a laptop running Windows 10 with an Intel i7-6700HQ 2.6GHz processor and 32 GB of memory.

We further evaluate the scalability of our approach and implementation to handle prints of complex models with many sub-parts. Numeric evaluation is found in Table 1. $\mathcal{B}$ denotes the set of voxels in the bounding box of the print. We observe that the distance computation ranges from a low of 67 ns per voxel for the dragon to a high of 340 ns per voxel for the anatomy model, with the other models < 100 ns per voxel. The high computation time for the anatomy model is likely due to a combination of cache performance for the larger slices and the large number of sub-objects.

Our pipeline is implemented in standard C++14. We exploit parallelism of the CPU via multi-threading of some computations. Although many aspects could be ported to GPU, we leave this for future work. One exception is that we use OpenGL and GLSL shaders to sample textures and assign RGBA values to surface voxels. The start-up delay for the anatomy model reflects an inefficiency in the (CPU) implementation when rasterizing many different sub-objects.

6.1 Color and translucency accuracy

In this section we evaluate the accuracy of our profile and pipeline in reproducing in-gamut color and translucency. We seek to validate two things: color stability, as in hue preservation, under decreasing α w.r.t. opaque, and accurate reproduction of α. To this end we printed gamut-mapped Color Checkers (CCs) with different α, shown in Fig. 7. Both color and translucency values were first gamut mapped. For color values, this means standard gamut mapping using absolute colorimetric intent into the opaque gamut. For α, this requires considering the color as well, since not all α are possible for all colors (*e.g.* fully transparent black).

Fig. 7 shows the printed CCs on a white background and Table 2 the color- and α-differences between the gamut mapped input and measured quantities. Fig. 8 shows close-ups of the CCs (top) lit from behind with a step function to show the change in background contrast w.r.t. α and (bottom) the two most opaque CCs under front lit conditions to show the edge-loss cue. The full CCs under backlit conditions can be seen in the supplemental appendix. Overall the color and translucency errors are very small and in a range which require mostly side-by-side comparison to see color shifts. There is one outlier, the transparent black patch (see first CC in Fig. 7), causing an unacceptable error of $\Delta E_{00} = 25.3$. We observe decreasing color accuracy for increasing translucency. This is caused by the opaque relative white point adjustments (Sec. 5.3) since chromatic components for white points corresponding to large γ are not zero. For small and large α the α-error rates are very small (below the anchor-pair difference in [Urban et al. 2017]) but by a factor of two larger for mid-range α.

Table 1. Performance details of our method.

Model	# sub-objects	# triangles	texture size (Mpixels)	Voxel Count		Compute Time		
				$|\mathcal{V}|$	$|\mathcal{B}|$	first chunk	total	distance
Anatomy (20 cm) Fig. 10	28	1.36×10^6	425	9.01×10^8	1.02×10^{10}	33min 28s	3hrs 57min	57min 49s
Head/brain (10 cm) Fig. 9	2	1.22×10^6	16.8	1.28×10^9	3.67×10^9	1min 9s	57min 53s	5min 26s
Dragon (10 cm) Fig. 14	1	7.21×10^6	-	2.02×10^8	3.17×10^9	1min 41s	26min 39s	3min 31s
Wax ship (10 cm) Fig. 13	1	1.0×10^6	37.75	2.65×10^8	6.36×10^8	25s	10min 11s	60s
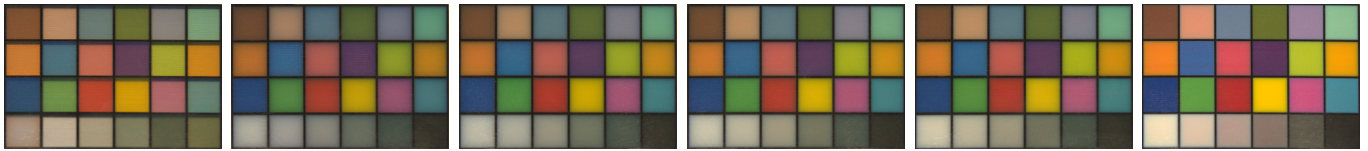
Fig. 7. Color checkers with varying degrees of translucency under front-lit conditions with a white backing. From left to right: $\alpha = 0.201$ (maximally transparent), $\alpha = 0.348$, $\alpha = 0.493$, $\alpha = 0.640$, $\alpha = 0.769$, $\alpha = 0.786$ (maximally opaque).
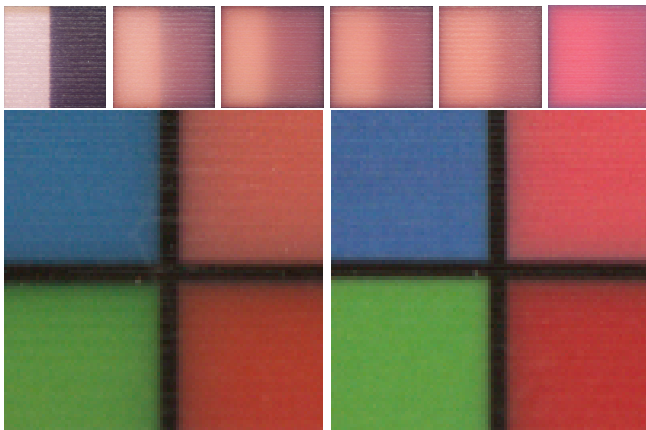
Fig. 8. Color checker close-ups. Top: lit from behind with a step function illuminant. The 3rd patch of the second row is shown. Same order as in Fig. 7. Bottom: front-lit close-up of CCs for $\alpha = 0.786$ (max. opaque, right) and $\alpha = 0.769$ (left). Note the increased blurring of the edges of the black cladding (1 mm thick) between patches on the left compared to the maximally opaque.

Table 2. Color accuracy in CIEDE2000 (ΔE) and translucency accuracy in $\Delta\alpha$ for 6 color checker prints of different input α values.

$CC(\alpha)$	ΔE_{median}	ΔE_σ	ΔE_{max}	$\Delta\alpha_{median}$	$\Delta\alpha_\sigma$	$\Delta\alpha_{max}$
CC(0.79)	2.3	1.7	6.2	0.06	0.02	0.10
CC(0.77)	3.2	1.9	8.6	0.02	0.02	0.08
CC(0.64)	3.0	1.9	8.0	0.06	0.03	0.13
CC(0.49)	3.7	2.2	9.1	0.15	0.06	0.24
CC(0.35)	4.2	2.4	8.8	0.23	0.09	0.29
CC(0.20)	5.5	4.8	25.3	0.07	0.08	0.29

Table 3. Euclidean hue errors (ΔH) in LAB2000HL [Lissner and Urban 2012]

ΔH_{median}	ΔH_{mean}	ΔH_σ	ΔH_{95ile}	ΔH_{max}
1.8	2.4	1.9	6.18	10.45

We noticed a magenta tinge in the gray ramp for the opaque CC (see last CC in Fig. 7). We believe this is caused by our gray component replacement (GCR= 0) strategy resulting in a minimum usage of black in the separation. This reduces graininess for bright colors but results in low color constancy and color mixture instabilities for grays. Changing this to GCR=100 would likely improve grays.

Note that the targets used to fit the optical printer model were printed $\approx$ 1.5 years earlier on a different machine than the CCs, i.e. the errors include also inter-machine and inter-material variability.

Since our gamut expansion strategy preserves hue up to the opaque relative white point adjustments (Sec. 5.3), we have also evaluated how well hue is preserved between the (not gamut mapped) input CC colors and measurements of the prints. Table 3 shows the ΔH error statistic computed in the nearly perceptually-uniform hue-linear LAB2000HL color space [Lissner and Urban 2012]. As expected the average errors are very small. The largest errors are in the bright blue patches of the transparent CC.

6.2 Design control of translucency for 3D color printing

We now show examples of how our approach allows 3D printing of objects with full color design of spatially-varying translucency, including both smooth translucency gradients and piecewise constant translucency. This can be used for aesthetics, or visualization/educational purpose by revealing complex internal structures
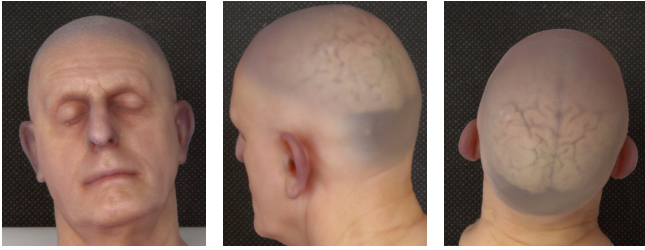
Fig. 9. An example of a smooth translucency gradient revealing an internal structure. Note that the skin color is not altered.
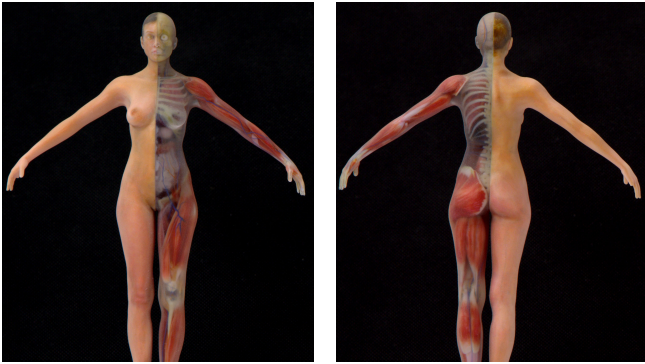

Fig. 10. An anatomy model printed with a skin sub-object of relative $\alpha = 0$ or $\alpha = 1$ based on the position within the print. The color of the skin was set to RGB=(1,1,1) for $\alpha = 0$.

Fig. 11. Head model printed with relative $\alpha = 1$ (left) and $\alpha = 0.5412$ (right). Identical model geometry and illumination conditions.

of objects in relation to their outer surface. All examples are easily created by standard editors, or by simple GLSL shaders.

Fig. 9 shows a head model, with a texture edited using ZBrush [Pixelogic 2017] to create a smooth α-gradient so that the back of the head is maximally transparent revealing a brain model.

Fig. 10 shows an anatomy model with a piecewise-constant α applied to the skin and skull using a shader. For the skin, α is set to 0 and RGB to $(1, 1, 1)$ if the x-coordinate (normalized to the extent of the print) is below 0.5, revealing muscles, bones, organs, arteries and veins. For the skull, α is set to 0 and RGB to $(1, 1, 1)$ if the

z-coordinate (again normalized) is greater than 0.4, revealing the brain. A close-up of the head is shown in Fig. 1 (a).

Fig. 11 shows a head model printed with maximum opaque α and the same model with an α of dominating lateral light transport. Both models have the same RGB texture. Skin wrinkles caused by surface topography and color texture are visible in the maximum opaque model whereas in the model dominated by lateral light transport they are strongly blurred. Note also the reduced lightness of the model with larger lateral light transport. This is caused by our opaque relative processing (Sec. 5.3) and a direct result of a white point with smaller lightness at the corresponding γ-level. Note that by encoding the transformation in a lookup-table we use a point process independent of spatial content aiming to ensure that the white points match for all γ-levels. This changes the lightness and chroma of reproducible colors for the benefit of preserving the relationship between colors and processing speed.

6.3 3D printing measured materials

We use the measurement setup described in [Urban et al. 2017] to obtain α values for a sample of the same wax used for the ship in Fig. 12 (right) and Fig. 14 (left). The wax ship was then scanned using an automated photogrammetric system, resulting in a 3D mesh and opaque color texture map. The results of fabricating the digital model are shown in Fig. 12 (left) and 13. Note that the prints have not been polished, and the resulting surface roughness reduces the apparent translucency compared to the smoother wax.

In Fig. 12 (left), the model is printed with the original color texture from the scan, and $\alpha = 0.6$ as measured from the wax sample. The print next to the original are illuminated from above with a directional light source. The color and translucency match quite well on a large scale. Differences are mostly due to small scale factors, such as surface roughness or air bubbles or other non-uniformities in the wax, which are difficult to model and beyond the scope of this paper. In Fig. 13, the model is printed with a constant color as measured from the wax sample, and varying α values. The prints are illuminated from below with an approximate point source, in addition to ambient daylight. The prints show a large range of translucency and changes in both amount and direction of subsurface light transport.

Fig. 14 shows how measured translucency appears under changes in geometry and color. On the left, we again see the wax ship, this time illuminated from behind by daylight. In the middle is a dragon printed with the material measurements for the wax (RGB and α), under the same illumination. The translucency and color match the wax very well. On the right, we have the same dragon with the same α, but the color is generated by a fractal noise function, resembling a mix of red and blue waxes. Again the translucency is highly plausible for wax. We see how the translucency is affected by color: blue areas absorb more light, reducing translucency.

7 LIMITATIONS

The results of our pipeline are naturally limited by the available materials, printer resolution, and other such hardware factors. The main limitation of the approach itself concerns alpha gradients covering the full range of α: Our approach to extend the printer's lateral light transport gamut by replacing white with clear material
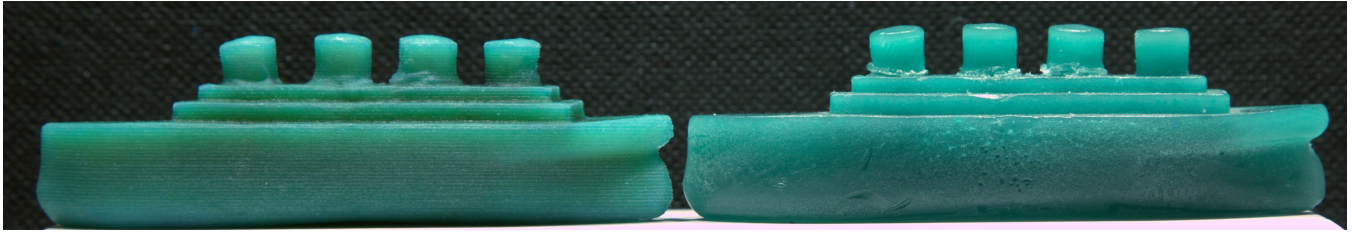
Fig. 12. 3D color and translucency copy (left) of a wax ship (right) illuminated with directional light from above.
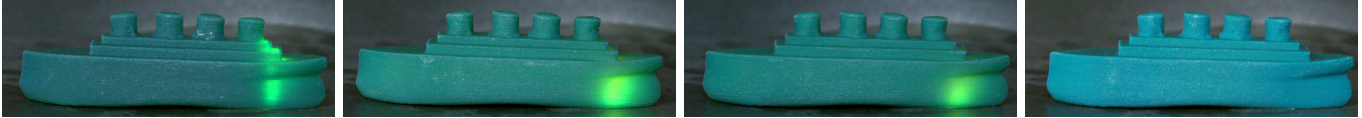

Fig. 13. 3D prints with color measured from a sample of the wax used to sculpt the ship in Fig. 12 illuminated with an approx. point source from below. From left to right: 3D print with relative $\alpha = 0, 0.1, 0.3, 1$ ($\alpha = 0.3$ corresponds to measured wax translucency of absolute $\alpha = 0.6$).

Fig. 14. From left to right, illuminated from behind by daylight: A ship figurine sculpted out of green wax, a 3D printed dragon with color and α measured from the same wax, a 3D printed dragon with the same α, but a fractal noise pattern for color, simulating a mix of red and blue waxes.
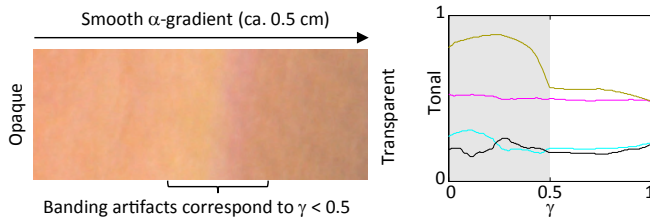

Fig. 15. Left: Banding artifacts in a spatially-short translucency gradient of a print. Right: CMYK tonal variations across γ to reproduce a constant opaque relative **CIELAB** color in the left gradient.

for small γ values only in the near surface region (Sec. 4.2.1) comes at with the cost that translucency metamers can only be reproduced by very different color tonal values. We observed this in α gradients that can only be reproduced by introducing large tonal gradients for constant color (see Fig. 15). If the lateral extent of such gradients on the surface is small, they may cause banding artifacts due to limitations in halftone resolution. A potential solution is to change step 4.2.1, possibly reducing the lateral light transport gamut.

Our choice of input limits the set of possible BSSRDFs. A 1D translucency space cannot model directional dependency, and making it a surface attribute reduces the degrees of freedom.

Another limitation is the large number of measurements required by our printer model to characterize the printing system. Improving the predictive power of the model to reduce the number of required measurements is an important direction for future work.

We also currently lack a practical soft-proof for the output of our pipeline. The color can be accurately predicted for any input, including spatially varying translucency, but the translucent appearance is not easily simulated, although α corresponds to absorption and scattering coefficients, which suggests a physically-based rendering approach should be feasible.

8 CONCLUSIONS

We have proposed techniques for multi-material 3D printing of spatially varying color and translucency via *translucency metamers*, which encode perceptual translucency cues, and presented efficient streaming algorithms to do so. The high costs of acquiring, processing and storing physical representations of BSSRDFs, the limited capacity of multi-material 3D printers to produce them, and the psychophysical results showing the human visual system senses translucency based on simple high-level cues, strongly motivate translucency metamers for joint color and translucency 3D printing.

Our approach facilitates the use of existing formats and 3D modeling software, while allowing the reproduction of an important part of the perceived translucency of measured materials. Combining

these through a library of α values for real materials is an interesting direction for future work.

ACKNOWLEDGMENTS

REFERENCES

3DSystems. 2014. Projet 860Pro. http://www.3dsystems.com/3d-printers/professional/projet-860pro. (2014).

Marc Alexa, Kristian Hildebrand, and Sylvain Lefebvre. 2017. Optimal Discrete Slicing. *ACM TOG* 36, 1, Article 12 (Jan. 2017), 16 pages. https://doi.org/10.1145/2999536

C. Arikan, A. Brunton, T. Tanksale, and P. Urban. 2015. Color-Managed 3D-Printing with highly Translucent Printing Materials. In *SPIE/IS&T Electronic Imaging Conference*. San Francisco.

V. Babaei, K. Vidimče, M. Foshey, A. Kaspar, P. Didyk, and W. Matusik. 2017. Color contoning for 3D printing. *ACM TOG (Proc. SIGGRAPH)* 36, 4 (2017), 124:1–124:15.

Barbieri. 2015. Spectro LFP qb. https://www.barbierielectronic.com/en/products/spectrophotometer/spectro-lfp-qb/91-869.html. (2015).

A. Brunton, C. Arikan, and P. Urban. 2015. Pushing the Limits of 3D Color Printing: Error Diffusion with Translucent Materials. *ACM TOG* 35, 1 (December 2015), 4:1–4:13.

Canon. 2014. Canon 5D Mark 3. http://www.usa.canon.com/cusa/professional/products/professional_cameras/digital_slr_cameras/eos_5d_mark_iii. (2014).

D. Chen, D.I.W. Levin, P. Didyk, P. Sitthi-Amorn, and W. Matusik. 2013. Spec2Fab: A Reducer-Tuner Model for Translating Specifications to 3D Prints. *ACM TOG (Proc. SIGGRAPH)* 32, 4 (2013).

M. Detrixhe, F. Gibou, and C. Min. 2013. A parallel fast sweeping method for the Eikonal equation. *J. Comput. Phys.* 237 (2013), 46–55.

Y. Dong, J. Wang, F. Pellacini, X. Tong, and B. Guo. 2010. Fabricating Spatially-Varying Subsurface Scattering. *ACM TOG (Proc. SIGGRAPH)* 29, 4 (2010).

O. Elek, D. Sumin, R. Zhang, T. Weyrich, K. Myszkowski, B. Bickel, A. Wilkie, and J. Křivánek. 2017. Scattering-aware Texture Reproduction for 3D Printing. *ACM TOG (Proc. of SIGGRAPH Asia)* 36, 6 (2017), 241:1–241:15.

M. D. Fairchild. 2005. *Color Appearance Models* (2 ed.). John Wiley & Sons, inc., West Sussex, England.

P.F. Felzenzwalb and D.P. Huttenlocher. 2012. Distance Transforms of Sampled Functions. *Theory of Computing* 8, 19 (2012), 415–428.

R. W. Fleming and H.H. Bülthoff. 2005. Low-level image cues in the perception of translucent materials. *ACM TAP* 2, 3 (2005), 346–382.

I. Gkioulekas, B. Xiao, S. Zhao, E. H. Adelson, T. Zickler, and K. Bala. 2013. Understanding the role of phase function in translucent appearance. *ACM TOG* 32, 5 (2013), 147:1–147:19.

M. Goesele, H. Lensch, J. Lang, C. Fuchs, and H.P. Seidel. 2004. DISCO: acquisition of translucent objects. In *ACM TOG (Proc. SIGGRAPH)*, Vol. 23. ACM, 835–844.

M. Hašan, M. Fuchs, W. Matusik, H. Pfister, and S. Rusinkiewicz. 2010. Physical Reproduction of Materials with Specified Subsurface Scattering. *ACM TOG (Proc. SIGGRAPH)* 29, 3 (2010).

HP. 2018. HP Jet Fusion. http://www8.hp.com/us/en/printers/3d-printers/3dcolorprint.html. (2018).

R. S. Hunter and R. W. Harold. 1987. *The Measurement of Appearance* (2 ed.). John Wiley Sons.

Image Engineering. 2015. CAL1 Calibration Light Source. https://www.image-engineering.de/products/equipment/illumination-devices/379-cal1. (2015).

H. W. Jensen, S. R. Marschner, M. Levoy, and P. Hanrahan. 2001. A practical model for subsurface light transport. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques (ACM-SIGGRAPH)*. ACM, 511–518.

M. W. Jones, J. A. Baerentzen, and M. Sramek. 2006. 3D Distance Fields: A Survey of Techniques and Applications. *IEEE TVCG* 12, 4 (2006).

P. Kubelka and F. Munk. 1931. Ein Beitrag zur Optik der Farbanstriche. *Zeitschrift für Technische Physik* 12 (1931), 593–601.

I. Lissner, J. Preiss, P. Urban, M. Scheller Lichtenauer, and P. Zolliker. 2013. Image-Difference Prediction: From Grayscale to Color. *IEEE TIP* 22, 2 (2013), 435–446.

I. Lissner and P. Urban. 2012. Toward a Unified Color Space for Perception-Based Image Processing. *IEEE TIP* 21, 3 (2012), 1153–1168.

F. Liu and Y.J. Kim. 2014. Exact and Adaptive Signed Distance Field Computation for Rigid and Deformable Models on GPUs. *IEEE TVCG* 20, 5 (2014), 714–725.

MCor Technologies. 2014. Iris. http://mcortechnologies.com/3d-printers/iris/. (2014).

P. Morovič, J. Morovič, I. Tastl, M. Gottwals, and G. Dispoto. 2017. HANS3D: A Multi-Material, Volumetric, Voxel-By-Voxel Content Processing Pipeline for Color and Beyond. *Color and Imaging Conference* 2017, 25 (2017), 219–225.

I. Motoyoshi. 2010. Highlight–shading relationship as a cue for the perception of translucent and transparent materials. *Journal of vision* 10, 9 (2010), 6–6.

F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. 1977. *Geometrical considerations and nomenclature for reflectance.* United States. National Bureau of Standards.

S. E. Palmer. 1999. *Vision science: Photons to phenomenology.* MIT press.

D. Panozzo, O. Diamanti, S. Paris, M. Tarini, E. Sorkine, and O. Sorkine-Hornung. 2015. Texture mapping real-world objects with hydrographics. *Computer Graphics Forum* 34, 5 (2015), 65–76.

M. Papas, C. Regg, W. Jarosz, B. Bickel, P. Jackson, W. Matusik, S. Marschner, and M. Gross. 2013. Fabricating Translucent Materials using Continuous Pigment Mixtures. *ACM TOG (Proc. SIGGRAPH)* 32, 4 (2013).

P. Peers, K. vom Berge, W. Matusik, R. Ramamoorthi, J. Lawrence, S. Rusinkiewicz, and P. Dutré. 2006. A compact factored representation of heterogeneous subsurface scattering. In *ACM TOG (Proc. SIGGRAPH)*, Vol. 25. ACM, 746–753.

T. Pereira, S. Rusinkiewicz, and W. Matusik. 2014. Computational Light Routing: 3D Printed Optical Fibers for Sensing and Display. *ACM TOG* 33, 3 (2014).

Pixelogic. 2017. ZBrush. http://pixologic.com/. (2017).

R. Rolleston and R. Balasubramanian. 1993. Accuracy of Various Types of Neugebauer Model. In *IS&T/SID*. Scottsdale Ariz., 32–36.

C. Schüller, D. Panozzo, A. Grundhöfer, H. Zimmer, E. Sorkine, and O. Sorkine-Hornung. 2016. Computational Thermoforming. *ACM TOG (Proc. SIGGRAPH)* 35, 4 (2016).

M. Schwarz and H.-P. Seidel. 2010. Fast Parallel Surface and Solid Voxelization on GPUs. *ACM TOG (Proc. SIGGRAPH Asia)* 29, 6 (2010).

J.A. Sethian. 1999. Fast Marching Methods. *SIAM Rev.* 41, 2 (1999), 199–235.

Y. Song, X. Tong, F. Pellacini, and P. Peers. 2009. SubEdit: A Representation for Editing Measured Heterogeneous Subsurface Scattering. *ACM TOG (Proc. SIGGRAPH)* (2009).

Stratasys. 2016. J750. http://www.stratasys.com/3d-printers/production-series/stratasys-j750. (2016).

Stratasys. 2017. GrabCAD Voxel Print. http://www.stratasys.com/software. (2017).

A. Sud, N. Govindaraju, R. Gayle, and D. Manocha. 2006. Interactive 3D Distance Field Computation using Linear Factorization. In *Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games (I3D)*. 117–124.

X. Tong, J. Wang, S. Lin, B. Guo, and H.Y. Shum. 2005. Modeling and rendering of quasi-homogeneous materials. In *ACM TOG (Proc. SIGGRAPH)*, Vol. 24. ACM, 1054–1061.

P. Urban, T. M. Tanksale, A. Brunton, B. Minh Vu, and S. Nakauchi. 2017. Redefining A in RGBA: Towards a Standard for Graphical 3D Printing. *CoRR* abs/1710.00546 (2017). arXiv:1710.00546 http://arxiv.org/abs/1710.00546

K. Vidimče, S.-P. Wang, J. Ragan-Kelley, and W. Matusik. 2013. OpenFab: A Programmable Pipeline for Multi-Material Fabrication. *ACM TOG (Proc. SIGGRAPH)* 32, 4 (2013).

Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. 2004. Image quality assessment: From error measurement to structural similarity. *IEEE TIP* 13, 4 (2004), 600–612.

K.D.D. Willis, E. Brockmeyer, S.E. Hudson, and I. Poupyrev. 2012. Printed Optics: 3D Printing of Embedded Optical Elements for Interactive Devices. In *ACM UIST*.

D. R. Wyble and R. S. Berns. 2000. A Critical Review of Spectral Models Applied to Binary Color Printing. *Color Research and Application* 25, 1 (2000), 4–19.

B. Xiao, B. Walter, I. Gkioulekas, T. Zickler, E. H. Adelson, and K. Bala. 2014. Looking against the light: How perception of translucency depends on lighting direction. *Journal of vision* 14, 3 (2014), 17–17.

XYZ Printing. 2017. Da Vinci Color. https://www.xyzprinting.com/en-GB/product/da-vinci-color. (2017).

K. Yoshida, N. Komeda, N. Ojima, and K. Iwata. 2011. Simple and effective method for measuring translucency using edge loss: optimization of measurement conditions and applications for skin. *Journal of biomedical optics* 16, 11 (2011), 117003–1170038.

H. Zhao. 2004. A Fast Sweeping Method for Eikonal Equations. *Math. Comp.* 74, 250 (2004), 603–627.

H. Zhao. 2007. Parallel Implementations of the Fast Sweeping Method. *Journal of Computational Mathematics* 25, 4 (2007), 421–429.