

Pushing the Limits of 3D Color Printing: Error Diffusion with Translucent Materials

Alan Brunton, Can Ates Arikan and Philipp Urban
Fraunhofer Institute for Computer Graphics Research IGD

Accurate color reproduction is important in many applications of 3D printing, from design prototypes to 3D color copies or portraits. Although full color is available via other technologies, multi-jet printers have greater potential for graphical 3D printing, in terms of reproducing complex appearance properties. However, to date these printers cannot produce full color, and doing so poses substantial technical challenges, from the sheer amount of data to the translucency of the available color materials. In this paper, we propose an error diffusion halftoning approach to achieve full color with multi-jet printers, which operates on multiple isosurfaces or layers within the object. We propose a novel traversal algorithm for voxel surfaces, which allows the transfer of existing error diffusion algorithms from 2D printing. The resulting prints show tremendous detail and realism.

Categories and Subject Descriptors:

General Terms: 3D color printing, half-toning, error diffusion

■

1. INTRODUCTION

In this paper, we present algorithms for producing 3D color prints, which are substantially more detailed and realistic than those produced by existing techniques. From a technical standpoint, we consider the problem of halftoning a signal defined on a surface for the purposes of accurate color reproduction in 3D printing when using translucent materials. More specifically, we present algorithms for error-diffusion on voxel representations of surfaces, which are compatible with translucent printing materials.

Accurate color reproduction is important in many applications of 3D printing, especially for design-prototypes or 3D color copies, where a texture-mapped 3D scan of an object is to be reproduced in a color-consistent way; this includes 3D portraits.

While 3D color printing has existed for some time using powder-binder [3DSystems 2014] and layer-laminated [MCor Technologies

2014] systems, multi-jet 3D printers with sufficient materials for color printing have only recently come on the market [Stratasys 2014]. In comparison to powder-based systems, multi-jet printers produce smoother surfaces at higher resolutions, and allow more control over the internal structure of the print. In the longer term, such printers have much greater potential to reproduce complex appearance properties. In fact, multi-jet printers with fewer materials have been used to approximate desired single-tone subsurface scattering properties [Hašan et al. 2010; Dong et al. 2010], and opaque objects inside transparent shells [Vidimčec et al. 2013].

Without accurate color reproduction, more complex appearance characteristics are limited in their applicability and appeal. Therefore, accurate color reproduction is a crucial step to fully exploiting the appearance capabilities of these printers. We provide precisely this critical building block for graphical 3D printing with this paper.

However, there are substantial technical challenges in color reproduction with multi-jet printers. First, is the amount of data that must be processed. For accurate color, the material assigned to each voxel must be controlled. Modern 3D printers have a resolution of more than 18 million voxels per cubic centimeter. Thus, prints such as those in Figure 1 require tens of billions of voxels (see Table I for details). Both resolution and build volume of 3D printers are expected to increase in the coming years.

Second, for multi-jet printers currently on the market, the colored materials are highly translucent, which means that the organization of the materials a significant distance beneath the surface, up to a few millimeters, can greatly affect the perceived color as discussed in Section 3. This translucency leads to both blurring of fine-scale details in textures, and to artifacts caused by severe dot-gain if care is not taken in the material arrangement. It is therefore highly important to control material placement several layers of voxels beneath the surface, considering the transmission and scattering properties of the materials, greatly complicating the computational aspects of halftoning algorithms.

While we expect less translucent materials to come on the market in the future, not only will our proposed technique also work for these materials, it will actually perform better—producing equal or larger color gamuts with less computational effort. Further, employing translucent materials may have its own advantages, when we consider materials with similar translucency as observed in skin.

Third, while powder-binder and layer-laminated systems, like 2D color printing, may combine up to 3 inks at a single surface location, multi-jet printing allows exactly one material per voxel.

In this paper, we leverage the knowledge of decades of research in color imaging, color management and 2D color printing, to maximize the quality and exploit the full capabilities of high-resolution multi-material 3D printers—and push their limits towards realism. To this end, we propose a geometry-adaptive error diffu-

contact: {alan.brunton,can.ates.arikan,philipp.urban}@igd.fraunhofer.de
web: www.cuttlefish.de

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© YYYY ACM 0730-0301/YYYY/15-ARTXXX \$10.00

DOI 10.1145/XXXXXXXX.YYYYYYY

<http://doi.acm.org/10.1145/XXXXXXXX.YYYYYYY>



Fig. 1. Color 3D prints computed with our software and printed with a multi-material printer show tremendous details and realism. Three of the apples and the thumb are not printed.

sion halftoning algorithm, which includes the following technical contributions:

- A traversal algorithm for voxel representations of surfaces, which maps 2D anisotropic error diffusion filters onto a surface in a consistently oriented way (Section 5) and requires only local information to do so.
- A layered halftoning algorithm, which combines the traversal algorithm with an arbitrary 2D error diffusion algorithm, and can adapt to the translucency of the materials or increase the color gamut by varying the number of layers (Section 6).

We do not attempt to approximate the full BSSRDF of an object, but rather present halftoning algorithms capable of reproducing an object’s albedo texture even when the printing materials are highly translucent, under non-pathological illumination conditions, such as a light source in contact with (as in Figure 2) or inside the printed object. Note that a high-quality halftone, such as those produced by our algorithms, is necessary, but not sufficient to accurately reproduce an object’s color. Additionally, one needs colorimetrically or spectrally accurate measurements of the object’s color, which are typically not provided by common 3D scanners, and an accurate optical printer model. In Section 7.2 we describe a fully empirical model for our printer. A more accurate model is left for future work.

2. RELATED WORK

Existing 3D color printing technologies. Powder-binder [3DSys-tems 2014] and layer-laminated [MCor Technologies 2014] systems provide full color solutions. In both cases, CMYK inks are applied to a white base material. Although these technologies are effective for color printing, they do not provide the resolution or smooth surfaces of multi-jet systems and are limited to nearly opaque materials. They therefore have less potential for future combinations of color with complex appearance properties.

Appearance reproduction and multi-material fabrication. Spec2Fab [Chen et al. 2013] is a general reducer-tuner framework for specification-driven digital fabrication, which allows textures to be replicated on a 3D print. They use an error diffusion optimization of material layerings, effectively a contone, with a uniform error filter (error is pushed equally to all neighbors). Although an important first step in texture-mapping for multi-material printers, it does not allow for anisotropic error diffusion filters, and the iterative optimization prohibits a streaming architecture. OpenFab [Vidimčec et al. 2013] is a programmable fabrication pipeline

for 3D printers, which uses in-slice 2D error diffusion dithering [Floyd and Steinberg 1976]. The authors observe that by dithering in 3D they could avoid streaks. Our approach treats the color signal where it is defined—on the surface—by mapping 2D filters into the tangent space of the surface. As discussed below, this allows us to colorimetrically characterize the 3D printer in a geometry-independent way.

Multi-material 3D Printing has been used to reproduce specified subsurface scattering properties [Hašan et al. 2010; Dong et al. 2010]. Fabrication of directional BRDFs for planar or near-planar surfaces has been done using multi-material printing [Lan et al. 2013] and photolithography [Levin et al. 2013].

Given the work that has already been done using multi-material 3D printing to reproduce complex appearance properties, the introduction of full color opens up tremendous possibilities for future appearance fabrication.

Tone reproduction in FDM prints. Some recent work has focused on the challenging task of improving tone reproduction in fused deposition modeling (FDM) printing. Hergel and Lefebvre [Hergel and Lefebvre 2014] optimize seam placement in multi-filament FDM prints to hide or reduce artifacts from changing filaments. Reiner et al. [Reiner et al. 2014] perform a type of halftoning for FDM printers while maintain long filament paths. Switching filament heads not only creates artifacts, but also increases print time. Both of these methods are specific to FDM printers.

3D Halftoning. 3D Halftoning has been applied to material composition using 3D error diffusion filters [Lou and Stucki 1998; Doubrovski et al. 2014] and 3D dispersed-dot dithering [Cho et al. 2003]. For color and appearance reproduction, 3D error diffusion is not appropriate because material assignments closer to the surface will have a greater influence on the appearance of the object than material assignments deeper within the object. Thus, a 3D error diffusion filter would have to adjust its orientation during traversal to account for this and maintain a consistent orientation with respect to the surface. An isotropic filter would produce similar artifacts as are observed with isotropic filters in 2D error diffusion. In contrast, our approach of halftoning on multiple offset surfaces within the object, in addition to the surface itself, results in a halftone that inherently accounts for the geometry of the surface. The relative influence of voxels at different depths from the surface is calibrated in an offline process and built into an International Color Consortium (ICC) profile. Such an offline color calibration process would be very challenging for a 3D filter, because it would require calibrating every possible surface orientation.

2D Halftoning. Generations of researchers in the field of 2D

halftoning focused on finding methodologies to optimally arrange printed dots for maximizing print quality (by preserving tone and structure and shifting quantization errors to the highest spatial frequencies possible - see Section 3.2) subject to technical limitations of printing systems (e.g. the ability to accurately deposit isolated dots) [Lau and Arce 2001].

One category of algorithms, called *point processes*, allow a very fast computation of the halftone screens by thresholding pixels using a precomputed threshold mask that is tiled over the 2D image. The traditional clustered-dot order dithering to create amplitude modulated screens and dispersed-dot ordered dithering [Bayer 1973] fall into this category. The latter technique was adapted to 3D printing [Cho et al. 2003] accounting particularly for dot placement limitations of binder-jetting systems. One drawback of dispersed-dot ordered dithering is that frequency components given by the screen period are visible resulting in cross-hatch pattern artifacts. Avoiding such artifacts in point process techniques, requires large threshold masks - e.g. blue-noise masks [Mitsa and Parker 1992] or green-noise masks [Lau et al. 1999] - which are heavily distorted if applied on surface manifolds with non-zero Gaussian curvature.

A second category of algorithms, called *neighborhood processes*, considers a local pixel neighborhood for thresholding. Error-diffusion methods (see Appendix) belong to this category and were pioneered by Floyd-Steinberg [Floyd and Steinberg 1976]. Since then, the first error diffusion techniques (affected by visual artifacts from low-frequency structural patterns) were improved drastically producing visually pleasing halftones without artifacts. Multiple modifications have been proposed including edge enhancement [Eschbach and Knox 1991], tone-dependent diffusion filters [Ostromoukhov 2001; Li and Allebach 2004], threshold modulation [Zhou and Fang 2003] structure preservation [Chang et al. 2009] or memory efficient processing [Chang and Allebach 2003].

A last category employs models of the human visual system to optimize halftone pattern iteratively [Agar and Allebach 2005; Pang et al. 2008]. Such methods can produce the highest halftone quality, but are in general computationally much more expensive than methods belonging to the other categories. Furthermore, these methods iterate over the whole data and require substantial modifications for adapting them into a streaming architecture. Since iterative methods aim to preserve local structure, their quality advantages over other methods diminish with increasing print resolution.

Due to the low computational effort and high quality of 2D error diffusion achieved with small diffusion filters, we decided to adapt it to 3D color printing. One prior work addresses error diffusion on a surface [Alexa and Kyprianidis 2015]. This approach operates on meshes and traverses the vertices based on the availability of subsequent moves or neighbors to diffuse error to. While this approach could be applied to any graph structure, including voxels, it is not clear that it can be applied in a streaming architecture. To the best of our knowledge, we are the first to consider error diffusion halftoning in the context of both non-Euclidean domains and highly translucent materials. Moreover, we are the first to propose such a technique demonstrated to be applicable in practice to tens of billions of elements.

3. BACKGROUND

3.1 Light scattering and absorption in printing materials

Given an arrangement of multiple non-fluorescent printing materials with similar refractive indexes within a shape \mathcal{S} , light propaga-

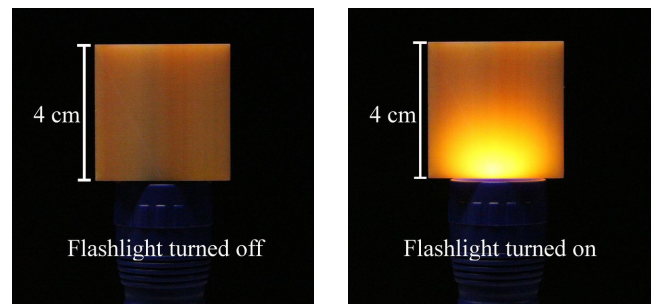


Fig. 2. Subsurface light scattering in photo-polymer printing materials.

tion within this shape can be described by the steady-state radiative transfer equation [Chandrasekhar 1960] (see supplemental material for details), which shows that the intensity of light traveling through the material is attenuated by absorption (dependent on wavelength, independent of direction) and is redistributed by scattering. Thus, a fraction of light entering the print at one location may be emitted from the surface at a different location due to scattering (see Figure 2). If light travels through different materials its spectral power distribution is modulated by each material's absorption coefficients and the path length within this material.

This has multiple implications for arranging translucent (high scattering, low absorption) printing materials colored in cyan (C), magenta (M), yellow (Y) and white (W) for full color 3D printing.

Highest Reflectance (white point): To maximize reflectance, not only the surface must be covered with white material, but also the space beneath. A significant fraction of light travels deep into the object due to the low absorption. Each non-white material with higher absorption placed into the light path would absorb light that would otherwise contribute to the albedo. There exists a minimum thickness d_w of a white layer to create almost maximum reflectance, i.e. for each $d > d_w$: $\|R_{d_w} - R_d\|_2 < \Delta R_{\max}$, where R_{d_w} and R_d are spectral reflectances of white layers with thicknesses d_w and d , and ΔR_{\max} is the maximally acceptable reflectance difference.

Lowest Reflectance (black point): To minimize reflectance, the printing material arrangement must ensure that light in the whole visible wavelength range is maximally absorbed. This can only be achieved by appropriately arranging C, M, Y materials on the surface and beneath. The minimal layer thickness d_b of such an arrangement for creating almost minimal reflectance (deviating only by ΔR_{\max} from minimum) satisfies $d_b < d_w$ because the color materials' absorption coefficients are larger than that of white (assuming similar scattering properties of all materials).

Therefore, not all voxels in the shape addressable by the printer need to be considered. We can drastically reduce computation by filling all voxels within the shape by default with white (maximizing reflectance of the white point) and compute the arrangement of remaining materials only within d_b of the surface. Although material placement with a distance $d \in [d_b, d_w]$ from the surface may impact reflectance, it affects only very bright colors which are also reproducible by color halftones in upper layers. While theoretically one could derive d_b based on measurements of the material parameters, Section 6.1 shows how selecting a maximum distance $d_{\max} \leq d_b$ from the surface allows an empirical trade-off between the number of voxels that must be addressed and the achievable gamut volume.

Optical Dot Gain: Voxels occupied by a colored material and surrounded by white material appear larger because the spectral

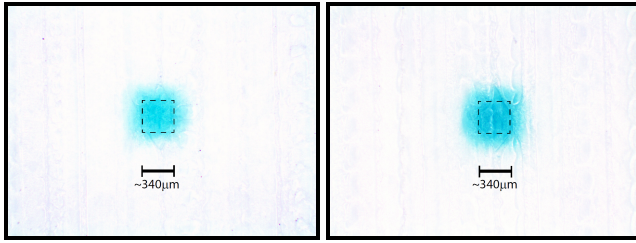


Fig. 3. Printing with translucent materials: severe dot gain and increased contrast by printing more slices. Two small cyan printed squares (indicated by a thin dashed line), both 8 dots or $\sim 340 \mu\text{m}$ wide at 600 DPI. Left: the square is printed 9 slices deep into surrounding white material. Right: 36 slices deep. Also visible is the surface topography due to printing roller.

power distribution of light traveling through both materials is modulated mostly by absorption of the colored material. This phenomenon is known in 2D printing as *optical dot gain* [Rogers 1997]. The more colored voxels are stacked on top of each other aligned with the surface normal the more laterally scattered light is modulated by the colored material. Thus, the colored area on the surface appears not only darker, but also larger as shown in Figure 3. Algorithms that use surface halftones to assign materials to inner voxels must consider this phenomenon to avoid dot gain artifacts particularly visible in bright areas by isolated disturbing large dots. Section 6.2 describes how independently halftoning layers allows to decorrelate material assignments in different layers, which greatly reduces dot gain artifacts in translucent materials.

3.2 Perceptual aspects in color printing

Each printing system is limited in reproducing reflectances due to the small number of available printing materials. Still, accurate color reproduction is possible by material arrangements creating physical errors for which the human visual system (HVS) is insensitive. Two perceptual aspects are particularly important in printing.

Contrast perception: Contrast sensitivity functions (CSFs) describe the HVS's sensitivity to achromatic and chromatic contrasts as a function of spatial frequency, orientation and luminance [Mullen 1985; Campbell et al. 1966; Van Nes and Bouman 1967]. The achromatic CSF has bandpass and chromatic CSFs have lowpass characteristics. Each CSF monotonically decreases for frequencies higher than 10 cycles/degree resulting in an apparent blurring of high frequency contrasts. This is used in printing by arranging printing materials in high frequency patterns (halftones) to create various color shades from only a few colored printing materials.

Color perception: Retinal processing reduces the information content of a spectral stimulus to only three values—the cone responses [Wyszecki and Stiles 2000]. Therefore, different reflectance spectra viewed under the same illuminant match visually if corresponding spectral stimuli yield similar cone responses. By agreeing on the viewing illuminant, accurate color printing is possible by material arrangements that reproduce cone responses instead of reflectances. A drawback of this *metameric color reproduction* is a likely mismatch between prints and originals for other illuminants. CIE colorimetry provides two standardized color spaces for metameric printing (see e.g. [Fairchild 2013]): CIEXYZ, which is linearly related to cone responses; and CIELAB, an opponent color space derived from CIEXYZ that allows simple access to color attribute predictors for lightness, chroma and hue and is perceptually more uniform. Color control in metameric printing is done by *separation*—relating CIEXYZ or CIELAB values to mate-

rial arrangements. Since this is generally impossible for all colors due to gamut limitations of the printing system, a preceding *gamut mapping* method must transform all colors into the device gamut aiming to minimize perceptual errors between original and reproduction [Morović 2008].

4. OVERVIEW

Given the resolution of current 3D printers, and expected increases in both resolution and build space, we need a streaming model for computation, wherein only localized data and computation are needed. Figure 4 shows such a streaming framework, which takes as input shape and color information, and produces printer compatible data in which a single material is assigned to each voxel. The error diffusion and material assignment algorithms described in Sections 5 and 6 are our core contributions and take place in the parts colored reddish. Each aspect of our framework is described briefly below.

Shape + Color: The input to our pipeline is a shape $\mathcal{S} \subset \mathbb{R}^3$ defined by a surface $\partial\mathcal{S}$ (e.g. tessellated mesh) with attached color information $\mathbf{f} : \partial\mathcal{S} \mapsto \mathcal{C}$ (texture data or per-vertex colors), where \mathcal{C} is specified by an invertible transformation from CIEXYZ defined for an illuminant (e.g. CIED50) and a set of color matching functions (CMFs) (e.g. CIE 1931 CMFs). Examples for \mathcal{C} are standard RGB color spaces [Süsstrunk et al. 1999]. By default we interpret RGB data as sRGB.

Voxelization: The axis-aligned bounding box of our shape $B(\mathcal{S})$ is discretized with a fixed, regular grid of voxels \mathcal{B} at the resolution given by the printer specification. We slice the surface and identify interior, $\mathcal{V} = \mathcal{S} \cap \mathcal{B}$, and surface voxels, $\mathcal{V}_\partial \subset \mathcal{V} = \partial\mathcal{S} \cap \mathcal{B}$, where a *slice* refers to a 2D array of voxels with a constant z -coordinate. The full slice is denoted by $\mathcal{B}(s) \subset \mathcal{B}$ for slice s with center z -coordinate z_s , and $\mathcal{V}_\partial(s) = \mathcal{V}_\partial \cap \mathcal{B}(s)$ denotes the surface voxels in slice s . Our voxelizer generates slices in chunks of ≈ 100 slices— $\approx 3 \text{ mm}$ for our hardware setup—depending on the number of layers and layer thickness as discussed in Section 6, until the object is completely sliced. Each chunk proceeds through the pipeline until materials have been assigned, and printer-specific output has been generated. Then the next chunk is generated.

During the voxelization process, we assign colors to the surface voxels; abusing notation slightly, we redefine $\mathbf{f} : \mathcal{V}_\partial \mapsto \mathcal{C}$ as a function on the surface voxels. We sample the texture data using trilinear interpolation (mipmapping) with the level-of-detail computed per surface voxel. This is key to avoid aliasing—while there are no viewpoint-dependent effects as in rendering, the texture map is in general non-uniform over the surface.

Color Management: In this stage, which includes gamut mapping and color separation, we map color data to printer tonal values using ICC color management [ICC 2010] (see section 7.2): $\mathbf{p} : \mathcal{C} \mapsto \mathcal{T}$, where each element of the tonal value space \mathcal{T} corresponds to the amount of available printing materials required to best reproduce a desired color. By function composition we attach one tonal value vector to each surface voxel: $\mathbf{g} = \mathbf{p} \circ \mathbf{f} : \mathcal{V}_\partial \mapsto \mathcal{T}$.

Layer Construction: To account for the translucency of the materials, see Figures 2 and 3, we transfer the tonal values stored in the surface voxels to the interior voxels within a distance d_{\max} of the surface. See Figure 8 for an example. These voxels have the greatest influence on the appearance of the printed object. From these voxels, we extract a set of layers as isosurfaces of distance to the surface of the object. The number and thickness of the layers determine d_{\max} , and these are chosen as described in Section 6.1.

Halftoning: We then treat each layer as a distinct surface with tonal values, and halftone them independently using the surface traversal

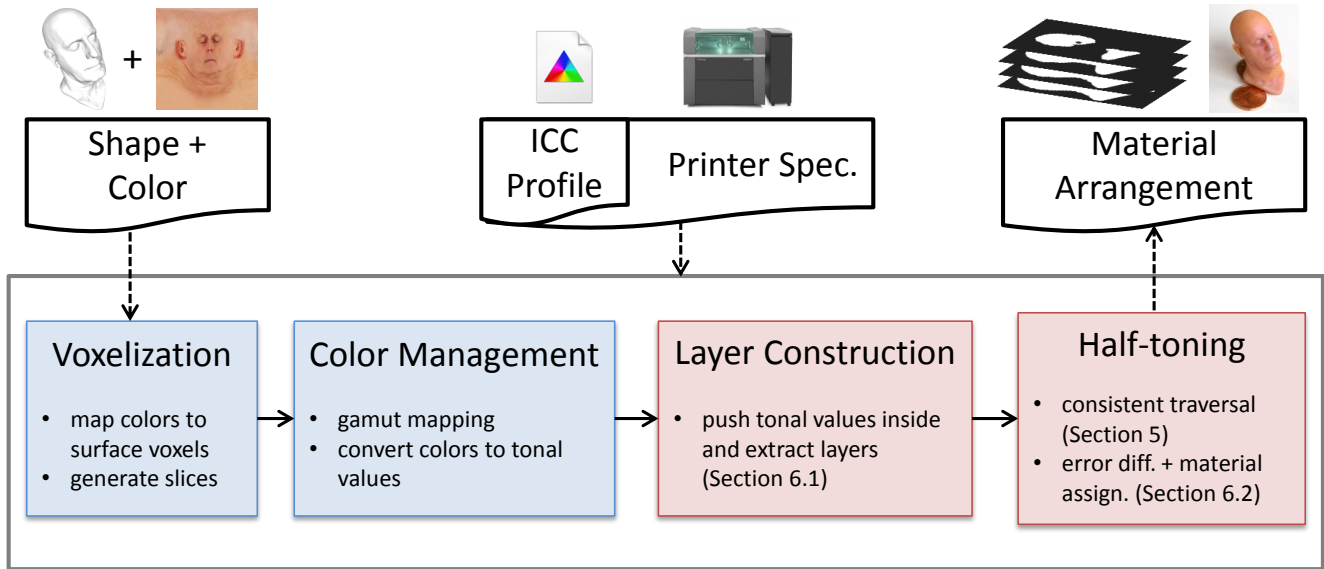


Fig. 4. Flow-chart of our processing pipeline.

algorithm described in Section 5 and the error diffusion and material assignment algorithm described in Section 6.

Material Arrangement: The output of the halftoning is a voxel-level material arrangement $m : \mathcal{V} \mapsto \mathcal{M}$, where the set of materials \mathcal{M} is discrete and small—4 in our case. We store m per-slice, and convert it to a printer-specific format to send to the device.

5. CONSISTENT SURFACE TRAVERSAL

Classical error diffusion algorithms for 2D images, e.g. [Floyd and Steinberg 1976; Ostromoukhov 2001], rely on the partial ordering property of subsets of Euclidean domains, to define a traversal order, eg. raster scan or serpentine scan order, in which error is never pushed to pixels that have already been halftoned.

On a surface, which has non-zero curvature and is in general not a topological disc, such a partial ordering property does not exist without introducing a seam or a singularity, a consequence of the Hairy Ball Theorem. Any traversal scheme will eventually encounter an element that has already been halftoned, requiring either a stop and restart, or a change in direction. Excessive stopping and restarting can lead to poor distribution of error, and we design a traversal that avoids this.

To map anisotropic error diffusion filters on to the surface requires a local coordinate system of the tangent plane at a given point, which is fixed with the surface normal and one additional direction. Mapping a filter to the surface so that it is consistently oriented for adjacent points is equivalent to parameterizing the tangent space of a surface in a consistent way, or finding smooth vector fields on a surface, which is in general an ill-posed problem and finding an optimal solution requires considering the entire surface.

Instead, we propose a voxel-level traversal algorithm that maintains a consistent orientation of the filter, traverses the surface voxels in long runs allowing error to accumulate and diffuse evenly, and does so using only local information. Our traversal avoids situations where voxels only distribute or obtain error during diffusion.

We maintain a consistent orientation of the filter by traversing the voxels in a consistent direction, and using the traversal direction to

fix a coordinate system in the tangent plane. While our traversal provides no theoretical guarantees, it performs well in practice.

5.1 Geometry of Voxel Slices

As a consistent direction, we always travel clockwise or counterclockwise about a line parallel with the positive z -axis. For brevity, we describe only the counterclockwise case. Note that we want to traverse counterclockwise, not about a global axis, but rather about a line through the center of the local *connected component* of voxels. Depending on the geometry of \mathcal{S} , when it intersects a slice plane $z = z_s$ and is discretized into a voxel representation, the resulting voxels in slice s may belong to multiple connected components. An example of a slice with multiple connected components can be seen in Figure 8.

If the surface normal is close to vertical (almost parallel with the positive or negative z -axis), then $\mathcal{V}_\partial(s)$ will be multiple voxels wide. These surface voxels $\mathcal{V}_\partial(s)$ may enclose interior voxels in slice s , and will be themselves enclosed by exterior or empty voxels in slice s . Figure 5 shows a 2D cross-section of this.

If we consider a part of the shape, which forms a connected component in consecutive slices, and where the surface faces almost in the negative z direction, but slopes or curves slightly upward, we have the following situation, which is shown in 2D in Figure 5. When projected into the xy -plane, the surface voxels of $\mathcal{V}_\partial(s)$ will be located inside the surface voxels of $\mathcal{V}_\partial(s+1)$. If we consider two surface voxels within the same connected component in slice s , the innermost should be traversed first, since they are next to $\mathcal{V}_\partial(s-1)$ and have already received error diffused from slice $s-1$. The outermost should be traversed last, which are next to $\mathcal{V}_\partial(s+1)$. The situation is reversed for upward facing surfaces.

5.2 Traversal Algorithm

We construct an undirected graph for traversal as follows: for each surface voxel $\mathbf{v} \in \mathcal{V}_\partial$, we connect it with an edge to each other surface voxel within the $3 \times 3 \times 3$ voxel window $\mathcal{W}(\mathbf{v})$ centered on \mathbf{v} . We denote this set of one-ring neighbors $\mathcal{N}(\mathbf{v}) \subset \mathcal{V}_\partial$. We traverse

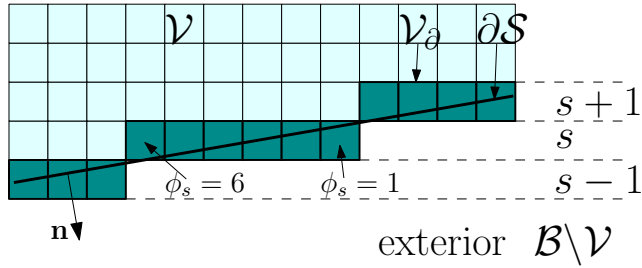


Fig. 5. Geometry of surface voxels in consecutive slices for a down-facing surface.

the voxel representation of the surface slice by slice, analogous to halftoning an image row by row, by selecting the next voxel from the set of neighbors that are within the same slice, $\mathcal{N}_s(\mathbf{v}) \subseteq \mathcal{N}(\mathbf{v})$. Error is always diffused upwards to the next slice, and to other surface voxels within the slice, which have not yet been assigned materials, as shown for a very small example (1 cm) in Figure 6.

One might also consider standard graph traversals, such as breadth-first or depth-first, also constrained to neighbors within the same slice, but these do not maintain a consistent direction of travel and therefore only permit symmetric error diffusion filters (e.g. uniform, Gaussian, etc.).

As described in Section 5.1, the traversal winds up the surface on a slice-by-slice basis counterclockwise about a line through the center of each connected component. However, finding the center of the current connected component would require a search over all voxels in the slice to determine the connected components, find their centers, and to which connected component each surface voxel belongs. Such a search would prove too expensive—essentially involving a pre-traversal.

We can determine whether a potential next voxel in proper traversal order winds about the connected component in the correct direction using only two pieces of local information. The first is the surface normal. The second is the in-slice distance to the exterior of \mathcal{S} , or distance-to-empty, denoted by $\phi_s(\mathbf{v})$ for voxel $\mathbf{v} \in \mathcal{V}_\partial(s)$

$$\phi_s(\mathbf{v}) = \min_{\mathbf{u} \in \mathcal{B}(s) \setminus \mathcal{V}(s)} \|\mathbf{v} - \mathbf{u}\|_1, \quad (1)$$

which can be pre-computed for each slice—see Figure 5 for an example. We use the L_1 -norm for simplicity and efficiency, but another norm would also work. The gradient of $\phi_s(\mathbf{v})$ always points to the interior of the connected component to which \mathbf{v} belongs. This is important for situations where the current slice may have multiple disconnected components, as in Figure 8.

This allows us to efficiently test if a potential next voxel $\mathbf{u} \in \mathcal{N}_s(\mathbf{v})$ continues in the same direction around the connected component by computing $(\mathbf{u} - \mathbf{v}) \times \nabla \phi_s(\mathbf{v})$. For example, if we are traversing counterclockwise, we can rule out all neighbors of \mathbf{v} for which this cross-product has a negative z -component. We denote the subset of $\mathcal{N}_s(\mathbf{v})$ satisfying this direction criterion $\mathcal{N}_\times(\mathbf{v})$.

Referring to Figure 5 and recalling the observations of Section 5.1 we get the following additional traversal criterion. If the surface faces downward, we wish to stay as *inside*—away from exterior voxels—as possible. If the surface faces upward, we wish to stay as *outside* as possible. Hence we choose the neighbor

$$\mathbf{w} = \arg \max_{\mathbf{u} \in \mathcal{N}_\times(\mathbf{v})} \phi_s(\mathbf{u}) \quad (2)$$

for down-facing surfaces, and

$$\mathbf{w} = \arg \min_{\mathbf{u} \in \mathcal{N}_\times(\mathbf{v})} \phi_s(\mathbf{u}) \quad (3)$$

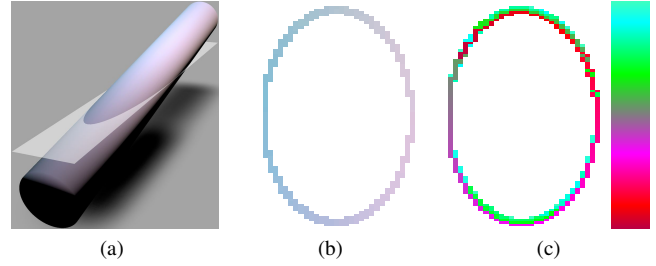


Fig. 7. A visualization of traversal order for a slice containing both upward and downward facing parts: (a) slice plane intersecting the object; (b) tonal values at the surface in that slice; (c) color-coded traversal order with color-coding legend (bottom=first, top=last).

otherwise.

Because the surface orientation is determined locally using the surface normal, the traversal adjusts as it traverses a slice with both up- and down-facing segments. Figure 7 shows the traversal order for such a slice.

5.3 Boundary Cases

Start point selection. We also use the distance-to-empty, in combination with one other local quantity, to select the start point in each slice. First, potential starting voxels (voxels in the slice, which have not yet been traversed) are filtered based on their *error diffusion count*, or number of times they have had error diffused to them from previous slices. We select the subset of voxels with the largest error diffusion count, and subsequently select from this subset using the distance-to-empty. If the z -component of the surface normal is negative, the start point is the point, which maximizes the distance-to-empty. If the z -component of the surface normal is positive, the start point is the point, which minimizes the distance-to-empty (i.e. one of outer-most points).

Once the start point and traversal direction are selected, the direction is tested to ensure it is not a dead end. If no voxel is available in that direction, the direction is reversed. This reduces the occurrence of single voxels with few neighbors to push error.

Birth and death of components. During the slicing process, new parts of the shape are encountered, creating new connected components of voxels in a given slice. When a new component is encountered in a given slice, the surface voxels will form a disc in that slice, meaning it can be traversed like an image. We handle these cases specifically as follows. Rather than spiral out, as with other downward facing surfaces, we select a global axis, x or y , and perform a 2D serpentine scan in this direction. This avoids situations where a long strip of voxels mostly has error pushed away from it, creating a start-up artifact that would be noticeable for low tonal values. We also use a 2D serpentine traversal for components, which are ending in the current slice, rather than spiraling in.

We detect the birth of a component when all possible starting points have an error diffusion count of 0, and the death of a component when all possible starting points have no neighbors in slice $s + 1$.

Serpentine Surface Scan. When we can no longer traverse in the same direction locally (all $\mathbf{u} \in \mathcal{N}_\times(\mathbf{v})$ have been traversed), we reverse direction, as in the serpentine traversal scheme from 2D error diffusion, which is shown to provide significant improvement over raster-scan traversal in images [Ulichney 1987; Chang and Allebach 2003]. Further, when selecting a new start point, we set the traversal direction opposite the traversal direction, in which error was last diffused to the new starting voxel.

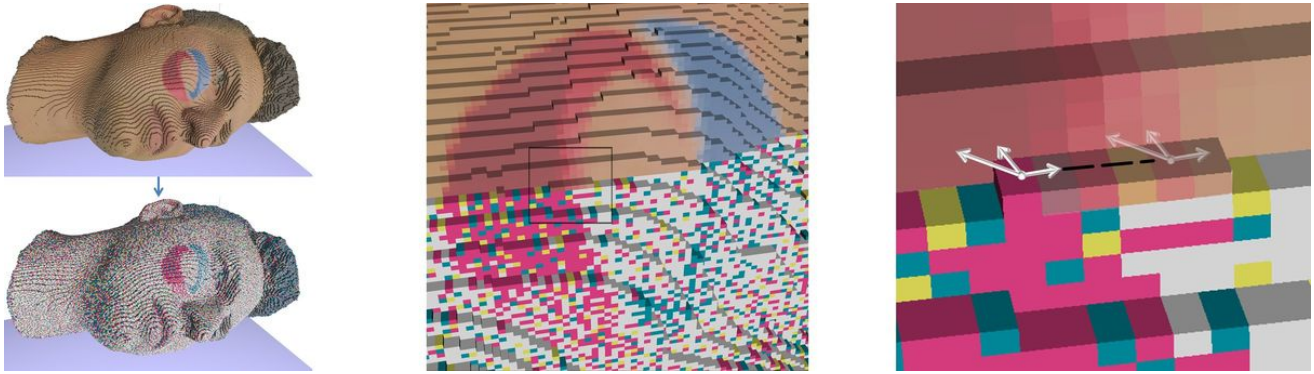


Fig. 6. Traversal of a surface maintaining consistent orientation of an error diffusion filter. Left: a rendering of a 1 cm voxelization of a head model before and after halftoning. Middle: a close-up of the halftoning process up to a given slice. Right: a cut-out from the middle of how a 3-component filter moves as each voxel is quantized. Note: the halftone is shown as opaque and diffuse for illustration purposes, and does not reflect the printed appearance.

5.4 Mapping the filter

Once we have chosen the next voxel to traverse, we can establish a local coordinate system at the current surface voxel, with which we can align the error diffusion filter and the neighbors of the current voxel. As in image error diffusion, one non-zero filter element is aligned with the next voxel to be traversed. This gives us a direction, which together with the surface normal, gives us the necessary coordinate system for the tangent plane at each surface voxel. Depending on whether the direction is clockwise or counterclockwise, we use either a left-handed or right-handed coordinate system. Figure 6 shows how an error diffusion filter with 3 non-zero elements is mapped to the neighbors in the tangent space of a voxel as it traverses the surface voxels.

This is closely related to decal mapping methods such as discrete exponential maps [Schmidt et al. 2006], and to parallel transport on manifolds, and for large decals exponential maps provide a more general solution. However, the filters used for error diffusion are typically small enough that the neighbors can simply be projected into the tangent plane while preserving distances.

We let the error diffusion filter live in the local tangent coordinate system, orthogonally project the neighbors onto the tangent space, and find the best alignment of neighbors and non-zero filter elements. We align neighbors with filter elements using symmetric closest points to avoid a single neighbor being assigned to multiple filter elements. That is, both the filter element and the neighbor must be closest to each other.

5.5 Analysis

The distance to empty can be computed in time linear in the number of voxels in each slice using a distance transform [Felzenszwalb and Huttenlocher 2004]. While this means over the whole print job, every voxel in \mathcal{B} must be visited, the constant hidden in the $O(\cdot)$ notation is very small and the distance transform is highly efficient. Selection of the starting point requires a search over all unprocessed surface voxels, but for non-pathological cases this will only be necessary a small number of times per slice. Each traversal operation requires a constant number of operations, as the number of neighbors is constant. Normals can be computed in $O(|\mathcal{V}_\partial(s)|)$ per slice using a signed distance field as an implicit surface definition, as described in Section 6.1. Thus, the total time complexity to fully process \mathcal{S} is linear in the total number of voxels $O(|\mathcal{B}|)$ with a small constant, which is very well suited to a streaming architecture.

6. LAYERED HALFTONING

Applying a halftone only to the surface voxels will result in a very low contrast print, due to the translucency of the materials. We can see this from Figure 3, which shows how increasing the depth of color material assignment increases the contrast. In Figure 9, we see how increasing the depth of color assignment increases the color gamut. For this reason, we introduce *layered halftoning*, in which color data is transferred from the surface to additional iso-surfaces, or layers, within the object, which are then independently halftoned using the traversal scheme presented in Section 5 and an adaptive filter [Ostromoukhov 2001] and threshold error diffusion algorithm [Zhou and Fang 2003].

6.1 Layer Construction

Layered halftoning begins by transferring the surface tonal values \mathbf{g} from \mathcal{V}_∂ to a subset of \mathcal{V} within a distance d_{\max} of the surface. All interior voxels more distant from the surface are assigned a white material to maximize reflectance as explained in Section 3. This requires both many distance computations and a transfer operator $\mathbf{T} : \mathcal{V}_\partial \times \mathcal{T} \mapsto \mathcal{V} \times \mathcal{T}$, which maps functions on \mathcal{V}_∂ to functions on \mathcal{V} .

To compute the distance field $d : \mathcal{B} \mapsto \mathbb{R}^+$ of voxels to the surface, we leverage the fact that we are only interested in voxels within a given distance d_{\max} . This allows us to construct a 3D mask containing these distances at the printer resolution. The distance field $d(\mathbf{v})$ is initialized with a large value $d_{\text{null}} > d_{\max}$. Distance computation then proceeds by moving this mask over the surface voxels \mathcal{V}_∂ , and writing the mask value to the voxel at the corresponding offset, if that value is less than the value already there. While this approach is efficient only for relatively small distances, it has the advantage of being embarrassingly parallel and requiring only a less-than operation at each voxel.

As a transfer operator, we simply take the tonal value $\mathbf{g}(\mathbf{u})$ of the nearest surface voxel \mathbf{u} , allowing the transfer operator to piggy-back on the distance computation at minimal extra cost, i.e. interior tonal values are assigned as

$$\hat{\mathbf{g}}(\mathbf{v}) = \mathbf{g}(\mathbf{u}) \text{ such that } \mathbf{u} = \arg \min_{\mathbf{s} \in \mathcal{V}_\partial} \|\mathbf{v} - \mathbf{s}\|_2 \quad (4)$$

for $\mathbf{v} \in \mathcal{V}$. Aside from being easy to compute, this has the benefit of preserving high-frequencies in \mathbf{g} . Figure 8 shows the process of assigning sRGB color values to surface voxels, converting them

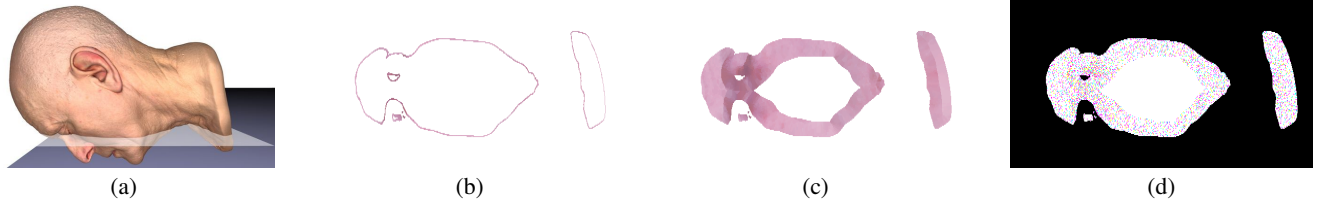


Fig. 8. Layered halftoning process: (a) slice plane $z = z_s$ intersecting the object; (b) the sRGB values in $\mathcal{V}_\partial(s)$ converted to tonal values \mathbf{g} via the map \mathbf{p} ; (c) transferred tonal values $\hat{\mathbf{g}}$ inside the object; and (d) material assignments m (black indicates no material).

to tonal values, transferring the tonal values to interior voxels and applying error diffusion to get the final material assignment. Note that because tonal values are pushed to the interior in 3D, interior tonal values may include surface tonal values from different slices.

Although there are inevitably discontinuities in $\hat{\mathbf{g}}$ at the cut loci of the distance field d (voxels \mathbf{v} where multiple surface voxels are equidistant), we found this not to be a problem as the tonal values mostly vary smoothly without significant correlation between high frequencies in \mathbf{g} and high curvature of the surface. In this case we simply assign tonals in a first-come first-served manner. Similarly, although areas with varying curvature result in slight tone shifts from \mathbf{g} to $\hat{\mathbf{g}}$, in particular for areas of high positive mean curvature, we did not find this to be a problem, even for small prints.

We extract layers of voxels with tonal values from \mathcal{V} by defining isosurfaces of d with which to segment \mathcal{V} . As isosurfaces we choose $d_\ell = \ell\tau$, where τ is the layer thickness, which we choose to be the voxel size along the dimension of minimum resolution. We use $\ell = 0, 1, \dots, L-1$, where L is an integer constant, which defines $d_{\max} = L\tau$. Ideally $d_{\max} = d_b$ (see Section 3.1) to minimize reflectance for maximizing contrast. However, we make a tradeoff between computational effort and gamut volume and set $L = 12$ for all prints shown in this paper. Figure 9 shows color gamuts computed using $L = 3, 6, 12, 18$ with the gamut volume as a fraction of the volume of sRGB. We can see how the volume increases with the number of layers. However, we found that the black point of the 18-layer profile, $L^* = 32.14$, is close to the minimum black point achievable with current CMYW materials— $L^* = 31.55$ for a 3 cm cube of full CMY mixture. Thus, little additional gamut is likely to be gained by more layers.

Voxel layer ℓ is then defined as the set of voxels between isosurfaces ℓ and $\ell + 1$,

$$\mathcal{V}_\ell = \{\mathbf{v} \in \mathcal{V} : d_\ell \leq d(\mathbf{v}) \text{ and } \exists \mathbf{u} \in \mathcal{W}(\mathbf{v}) : d(\mathbf{u}) < d_\ell\} \quad (5)$$

where the second condition is to ensure that the layers form thin voxel approximations of isosurfaces as much as possible, and \mathcal{W} is the voxel window defined in Section 5.2. Due to different resolutions along each axis, $d_{\ell+1} - d_\ell$ may be multiple voxels thick depending on the orientation. In the case of \mathcal{V}_0 , this condition is replaced by $\exists \mathbf{u} \in \mathcal{W}(\mathbf{v})$ such that $\mathbf{u} \notin \mathcal{V}$.

Note that we can very efficiently compute normals for all isosurfaces, required for traversal of the error diffusion filter (see Section 5.2): $\mathbf{n}(\mathbf{v}) = \nabla \tilde{d}(\mathbf{v})$, $\mathbf{v} \in \mathcal{V}_\ell$, where

$$\tilde{d}(\mathbf{u}) = \begin{cases} -d(\mathbf{u}) & \text{if } \mathbf{u} \in \mathcal{V} \\ d(\mathbf{u}) & \text{otherwise} \end{cases} \quad (6)$$

is a signed distance field, for $\mathbf{u} \in \mathcal{B}$. We use finite differences to compute $\nabla \tilde{d}$.

6.2 Error Diffusion and Material Assignment

We treat each voxel layer as a set of surface voxels, traverse them as described in Section 5, and halftone them as follows. Each tonal channel is halftoned independently using a standard error diffusion filter, mapped onto \mathcal{V}_ℓ as described in Section 5.4 and in the Appendix. At a given voxel $\mathbf{v} \in \mathcal{V}_\ell$, this results in a halftone vector $\mathbf{h}(\mathbf{v}) \in \{0, 1\}^T$ where $T = \dim(\mathcal{T})$. Thus, multiple materials may have a tonal value of 1 indicating they should be assigned to \mathbf{v} . Since we can only assign a single material $m(\mathbf{v})$ to each voxel, we apply the following tie-breaking scheme.

For each tonal channel $t \in \{1, \dots, T\}$, we maintain a tie-breaker error, $c(t)$, which is initialized to $c(t) = 0$ for all t at the start of each slice. If at a given voxel, one or more tonal values are halftoned to 1, the tonal channel $t^* = \arg \max_t c(t)$ with the largest tie-breaker error is declared the “winner” and we assign the corresponding material to the voxel. We then reset $c(t^*) = 0$ and increment $c(t)$ for all other t . If there is no t such that $\mathbf{h}(\mathbf{v})[t] = 1$, we assign white material to the voxel.

Note that this tie-breaking scheme is independent of the per-tonal error diffusion, and the error diffusion does not know when a tonal is quantized to 1, but the material is not assigned. This in general leads to a slight loss of tone, but is mitigated by the large number of voxels, and it is anyway accounted for by the color management, as described in Section 7.2, because we characterize the printer using targets printed with the same algorithm.

Note that errors are only diffused within the same layer, and each layer is halftoned independently. This has the following benefits. First, it allows different layers to be halftoned in parallel. Second, it further avoids the issue of resampling a halftoned signal, which would be necessary if the halftone was performed on the surface and then transferred to the interior. This is problematic because the halftoned signal is by design high-frequency and resampling it introduces artifacts. Third, we can use the threshold modulation of the halftoning algorithm [Zhou and Fang 2003] to avoid inter-layer correlations, which can create dot gain artifacts (see Section 3.1). We do this by seeding the pseudo-random number generators used for the threshold modulation differently for each slice and layer.

As discussed in Section 6.1, due to differences in resolution along the different axes, some interior voxels will fall in between layers, and not be directly halftoned. For these voxels, we simply assign the material of the nearest voxel that does belong to a layer.

7. EXPERIMENTS

7.1 Printing Setup

We use an Objet500 Connex3 from Stratasys [Stratasys 2014], for which the vendor has provided an interface allowing us to specify material assignments at the printer resolution. As the Connex3 allows only 3 model materials and 1 support material, we expanded

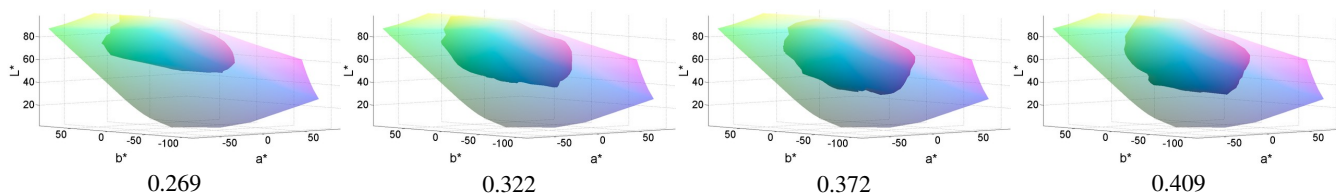


Fig. 9. Increasing number of layers results in increasing gamut volume. From left to right: 3, 6, 12 and 18 layers. The gamut volume is indicated below each image as a fraction of the volume of sRGB gamut shown transparent.

the possible gamut by coloring the support material with a yellow acid dye. Thus, we print with white, cyan, magenta and yellow materials. Care must be taken that there are no large agglomerations of support within the object, which would weaken the structural integrity of the print. Fortunately, as our halftone is by design high-frequency, this is avoided by scaling the yellow tonal value by a constant factor between 0 and 1. We found 0.3 to provide both a chromatic yellow and structurally stable prints. Our layer-based approach also allows us to prevent support from being present in the outermost layer (the surface), by applying a layer and tonal dependent soft-thresholding to the tonal values. For error diffusion we use a tone-adaptive filter [Ostromoukhov 2001] and threshold modulation technique [Zhou and Fang 2003].

7.2 Color Management

Specifying the color-to-tonal transformation \mathbf{p} requires at least two ICC profiles: An input profile that specifies texture-data colorimetrically and a printer profile that includes gamut mapping and color separation. Generating a printer profile involves a colorimetric characterization of the printer, i.e. we need to predict the printout’s CIEXYZ color for each tonal value vector in \mathcal{T} . For this, the tonal value space is sampled and the color of corresponding printouts is measured and used to fit a colorimetric printer model. We used a fully empirical model, which interpolates between color-measured printouts of densely sampled tonal values - for our CMYW printer we use a uniform $8 \times 8 \times 8$ sampling of the channel-wise linearized CMY tonal value cube. Linearization is performed employing the broadband Murray-Davies model according to [Wyble and Berns 2000]. This target is printed using the algorithm described in Sections 5 and 6. A picture of the resulting printout is shown in Figure 10 (left).

Measuring colors of such highly translucent printing materials is a challenging problem. It can be shown that measurements made by spectrophotometers used in the graphic arts community are systematically biased towards lower reflectance due to subsurface light transport away from the detection area. For our color measurements, we used a bidirectional $0^\circ/45^\circ$ measurement geometry, an almost colorimetric DSLR camera (Vora value of 0.9455 [Vora and Trussell 1993]), diffuse broadband illumination simulating CIED50, flat fielding (to account for optical path variations between light source, printed color patches and camera), and a polynomial approach to map camera RGB values to CIEXYZ. Color errors determined by spectroradiometric measurements are within the interinstrument-variability of spectrophotometers used in graphic arts for opaque materials. For more details, the reader is referred to [Arikan et al. 2015].

We evaluated the effectiveness of color management as follows. We took a standard color checker and measured each of the 24 patches with a spectrophotometer. We then mapped each of these colors into our printer gamut (12-layers, Figure 9, second from

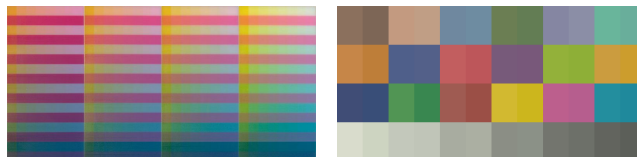


Fig. 10. Left: color characterization target with 12 layers. Right: predicted and measured color patches for a color checker; the left of each patch is the predicted color (gamut mapped) and the right is the measured color of the printed patch.

right), and printed planar patches. We then measured the printed patches with the almost colorimetric DSLR camera, and computed the CIEDE2000 differences to the gamut mapped colors—i.e. the colors predicted by our empirical printer model. We observed a median error of 2.2, a mean error of 2.3, and minimum and maximum errors of 0.5 and 6.1. The predicted and printed colors are shown in Figure 10 (right). Note that this evaluates *only the empirical printer model* and not the halftone, which is evaluated in Sections 7.3 and 7.4.

7.3 Evaluation

Performance. Our non-optimized implementation supports multi-threading for some computations (for example the transfer operator and halftoning each layer in parallel), but does not exploit any GPU capabilities, although several aspects would lend themselves well to GPU implementation. Table I shows performance data for several prints: the number of voxels, both those assigned a material ($|\mathcal{V}|$) and in the build volume ($|\mathcal{B}|$), computation times include the time to deliver the first chunk of slices to the printer and the total, and the print times. The results were collected on a standard desktop PC with an Intel Core i7-4770 processor and 32 GB of memory.

Tone Preservation. It is in general difficult to evaluate how well a 3D print reproduces the color of the input texture, due to the large change in appearance resulting from different illuminants and illumination distribution in combination with the curved surface. As discussed further in Section 7.6, the printing materials we use are highly color inconstant, meaning the perceived color changes under different illuminants. Further, typically the textured model is not rendered using the reflectance properties of the printing materials. In the case of scanned objects, evaluating with respect to the original object is meaningless, since the capture systems are typically not color calibrated.

Therefore, we evaluate the quality of tone preservation, rather than the similarity of perceived colors, as follows. We compute the root mean squared error (RMSE) between material assignments m and tonal values $\hat{\mathbf{g}}$, which are given for several models in the last column of Table I. First, we compute material fractions for each slice—the number of voxels assigned a given material in the

Table I. Performance details of our method.

Model	Voxel Count		Compute Time		Print Time	Material-Tonal RMSE			
	$ \mathcal{V} $	$ \mathcal{B} $	first chunk	total		C	M	Y	W
Bald head (20 cm) Figure 12d	8.6×10^9	2.49×10^{10}	1 min	6 h 10 min	21 h	0.0038	0.0074	0.0074	0.0184
Ruslan (15 cm) Figure 12a	5.7×10^9	1.36×10^{10}	1 min	4 h 18 min	11 h*	0.0097	0.0038	0.0107	0.0236
Ruslan (5 cm) Figure 12b	2.12×10^8	5.07×10^8	8 s	16 min	2 h	0.0094	0.0041	0.0120	0.0243
Nefertiti (20 cm) Figure 12e	5.0×10^9	1.67×10^{10}	1 min	4 h 55 min	20 h	0.0056	0.0041	0.0080	0.0173
Apple (8 cm) Figure 12c	1.8×10^9	3.7×10^9	30 s	1 h	10 h	0.0056	0.0030	0.0308	0.0247

See text for a description of how we compute the tonal errors. *Print job terminated after 69% of slices.

slice divided by the total number of voxels $\mathbf{v} \in \mathcal{V}(s)$ such that $d(\mathbf{v}) < d_{\max}$. Then we determine expected material fractions from the tonal values $\hat{\mathbf{g}}(\mathcal{V}(s))$ using the Demichel equations [Demichel 1924]. For example, we compute the expected material fractions for cyan and white from tonal values as

$$E_C = C(1 - M)(1 - Y) + \frac{1}{2}CM(1 - Y) + \frac{1}{2}C(1 - M)Y + \frac{1}{3}CMY \quad (7)$$

and

$$E_W = (1 - C)(1 - M)(1 - Y) \quad (8)$$

respectively, where C , M and Y are the average tonal values over the same set of voxels. The formulas are similar for magenta and yellow. Finally, we compute the errors as the difference between the expected material fraction and the actual material fraction for each tonal channel. We see that our algorithm preserves tone very well, with most errors being less than 1% of the tonal value range.

7.4 Visual Experiment

Visual experiments were conducted to assess the structural quality of the proposed halftone method, in particular the level of graininess and the visibility of structural artifacts. For this, we generated a test surface with the following function

$$z(x, y) = e^{-r^2/\sigma^2} (0.5 \cos(3r) + 0.5) \quad (9)$$

where $r = \sqrt{x^2 + y^2}$, $\sigma = 2R/3$, $R = 5$ cm with both x and y ranging from -5 cm to 5 cm. We then applied two textures to this surface, as shown in Figure 11; one with smooth color gradients and one with high-frequency patches; the colors were selected to uniformly cover the a^*b^* -plane of lightness $L^*=70$ and gamut-mapped using the absolute-colorimetric intent of our ICC profile.

The printed surfaces were placed on a colorimetrically characterized display next to their rendered versions (gap of approx. 1.25 cm - the left/right position of rendering and print was randomly chosen for each subject). We used an *Eizo ColorEdge cg301w* 30-inch display with a resolution of 2560×1600 pixels. The printed surfaces were 10 cm \times 10 cm, which covers 400×400 pixels on the screen. The screen was placed horizontally and illuminated from directly above using a *JUST NORMLICHT LED Color Viewing Light M* viewing booth at a distance of 1 m from the screen to provide a diffuse illumination. The illuminant used was LED-simulated CIED50. A chin-rest was placed so to obtain 60 pixels-per-degree (ppd) at the screen center (85 cm distance) and a 45° viewing angle to the screen. The renderings were done to approximately match this viewing condition and the luminance level of the print. Figure 11 shows pictures taken with an almost colorimetric

Table II. Results of the visual Experiments.

Experiment I (graininess level [noise std. in CIEDE2000])				
	mean	std	min	max
Smooth texture	0.9667	0.9994	0	4
Patch texture	2.0000	1.2594	0	4
Experiment II (structural artifacts [quality scale 0-5])				
	mean	std	min	max
Smooth texture	1.3000	0.7497	0	3
Patch texture	1.3667	0.8899	0	4

DSL camera of what the subjects viewed. A detailed image and diagram of the setup can be found in the supplemental material.

30 naive subjects with normal visual acuity – tested with the Snellen test – participated in the experiments (10 female and 20 male; average age of 30 years). For each of the two textures, a total of 10 distorted textures were created by adding zero-mean Gaussian pixel-noise with standard deviations of $1, \dots, 10$ CIEDE2000 units, simulating different levels of graininess. In a first experiment, subjects were asked to select the rendering with the distorted texture matching the graininess of the print. In a second experiment, they had to judge the level of structural artifacts in the print – showing the noise-free rendering as reference – on a quality scale of 0-5: 0 (not visible), 1 (visible but not disturbing), 2 (slightly disturbing), 3 (disturbing), 4 (very disturbing), 5 (extremely disturbing).

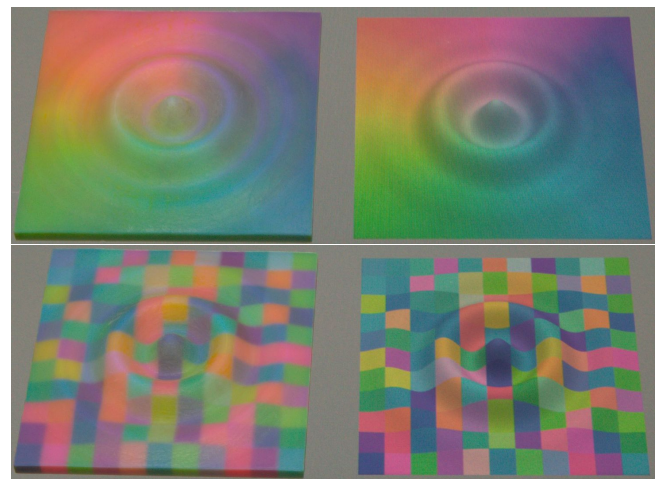


Fig. 11. Pictures of the test surfaces taken from the chin-rest position in our experiment; printed (left) and rendered (right). Moiré on the renderings is caused by the acquisition process.

Results of the experiments are shown in Table II. For both textures, the average perceived graininess level corresponds to a noise standard deviation smaller or equal to 2 CIEDE2000 units. To put this value into perspective, we computed the pixel-wise CIEDE2000 differences between the noise-free and distorted texture (noise std. of 2 CIEDE2000 units) using S-CIELAB with a visual resolution of 60 ppd [Johnson and Fairchild 2003]. The 99th percentile of CIEDE2000 errors is 1.644 (smooth texture) and 1.694 (patch texture). Since this is only slightly above the just noticeable distance, we can conclude that the perceived graininess of the halftone is very low. The second experiment revealed that also structural artifacts do not adversely affect the perceived quality of the halftone: for both textures, average quality scores are smaller than 1.4, i.e. artifacts are visible but not even slightly disturbing. Note that there are some drying related artifacts, which are likely considered in the quality scores but independent of the halftone.

7.5 Qualitative Results

Figure 12 shows some qualitative results demonstrating the detail and realism produced by our software using our hardware and material setup. Part (a) shows (left-to-right) a head scan with original texture rendered with diffuse shading, the same model with the texture gamut-mapped using our profile, and the resulting 3D print (15 cm tall). Note how well both smooth tones and high-frequency details are reproduced. Part (b) shows the same model printed 5 cm tall to show the effects of scale on the prints—note how similar the features of the two prints are despite the difference in scale. Part (c) shows a realistic-looking printed apple. Parts (d) and (e) demonstrate how well our halftoning algorithms reproduce high-frequency details in the textures. Prints (c)-(e) were generated with an error in the ICC profile, so the colors are off in absolute terms when compared to the textures. In part (d), the back of the ear is rendered next to the printed ear, which is about 3.5 cm high.

7.6 Limitations

Our setup is limited primarily from the hardware side. With four materials we are forced to print without a black. This means we have difficulty reproducing dark colors and a lack of color constancy—a change of illuminant alters the perceived color. We measured the color inconstancy index (CII) between CIED50 and CIEF11 of cyan, magenta, yellow and white, to be 4.65, 3.24, 5.28 and 0.59, respectively, using CAT02 for chromatic adaptation and CIEDE2000 for color difference [Fairchild 2013]. We also have no control over the translucency, as the materials have very similar scattering and transmission properties. We can only use their translucency to create realistic looking objects, not control it to a desired level. The introduction of printers with more materials, and new, more opaque materials would address these limitations, and open the possibility of combining color with desired scattering and reflective properties. A compelling open question is whether such complex appearance properties can be achieved using error diffusion-style techniques. A further limitation is due to the colored support material. During the uv-curing process, support material adjacent to model material mixes and binds with the model material. This results in a yellow tinge on some sloping and vertical surfaces. This could be overcome without additional materials by a multi-pass printing, in which the support material is cured separately from the other materials.

Algorithmically, our approach has the same limitations as 2D error diffusion methods: for extremely low tonal values, we encounter

mild start-up artifacts—materials are not placed entirely uniformly. We have found this only in extreme cases.

The curvature of the surface affects the transfer of tonal values to the interior layers when the mean absolute curvature H approaches $1/d_w$, where d_w is the depth of highest reflectance as discussed in Section 3; equivalently as the radius of curvature approaches d_w . High negative mean curvature (convex) regions may darken and high-frequency texture features that coincide with high positive mean curvature (concave) regions will “bloom”. We have not found this to be a major problem, although such effects show up strongest in small prints, e.g. the nose of the small Ruslan print in Figure 12 (b). We plan to address curvature-dependent effects in future work, e.g. by adjusting the tonal values according to curvature, which could be estimated using the signed distance field \tilde{d} in (6). The surfaces used in the visual experiment also demonstrate the extent to which curvature influences the perceived color. These surfaces, shown in Figure 11, contain both concave and convex regions with high curvature, and relatively flat regions on the periphery.

8. CONCLUSIONS

In this paper, we have presented algorithms for precise and efficient control of material placement in multi-jet 3D printers for the purposes of halftoning—a critical ingredient in accurate color reproduction. We have presented a layered halftone, which accounts for the high-translucency of currently available color materials, and a traversal algorithm for voxel representations of surfaces, which allows us to map 2D error diffusion filters onto a surface in a consistently oriented way. This allows us to leverage decades of knowledge from 2D error diffusion research. Our algorithms fit within an efficient streaming architecture, preserve tone and do not produce artifacts. We have further shown how our algorithms seamlessly allow the printer’s support material to be colored, thereby expanding the printer’s color gamut while preserving structural stability. The 3D color prints generated with our setup exhibit a high level of detail and realism.

Due to the ability to arbitrarily configure materials with different optical properties through an object, multi-jet printing has tremendous potential for graphical 3D printing in terms of reproducing complex appearance properties. By providing the first full color capabilities for these printers, we have provided a critical building block for graphical 3D printing.

The introduction of more opaque color materials will allow our algorithms to print with larger color gamuts using fewer layers, resulting in a performance boost. Better exploiting parallelism in our computations is a key direction for future work as printer resolutions and build volumes increase. As the number of printing materials increase, it will be important to consider printer models in characterizing the printers and computing profiles.

Acknowledgements

We give a big shout-out to Tejas Tanksale (TT) for his help in generating ICC profiles and conducting measurements. We thank Christoph Godau for access to the equipment at the IDD of TU Darmstadt. We thank Stratasys for providing the voxel-level printing interface to the printer. We thank Pedro Santos and the Cult-Lab3D of Fraunhofer IGD for making the Nefertiti buste available to us. We thank Christian Zeh from Clariant Produkte GmbH for coloring the support material. We thank Ten24 for making the head model available, and ScanLab and TurboSquid for making the Ruslan scan available. This work was funded by the FhG Internal Programs under Grant No. Attract 008-600075.

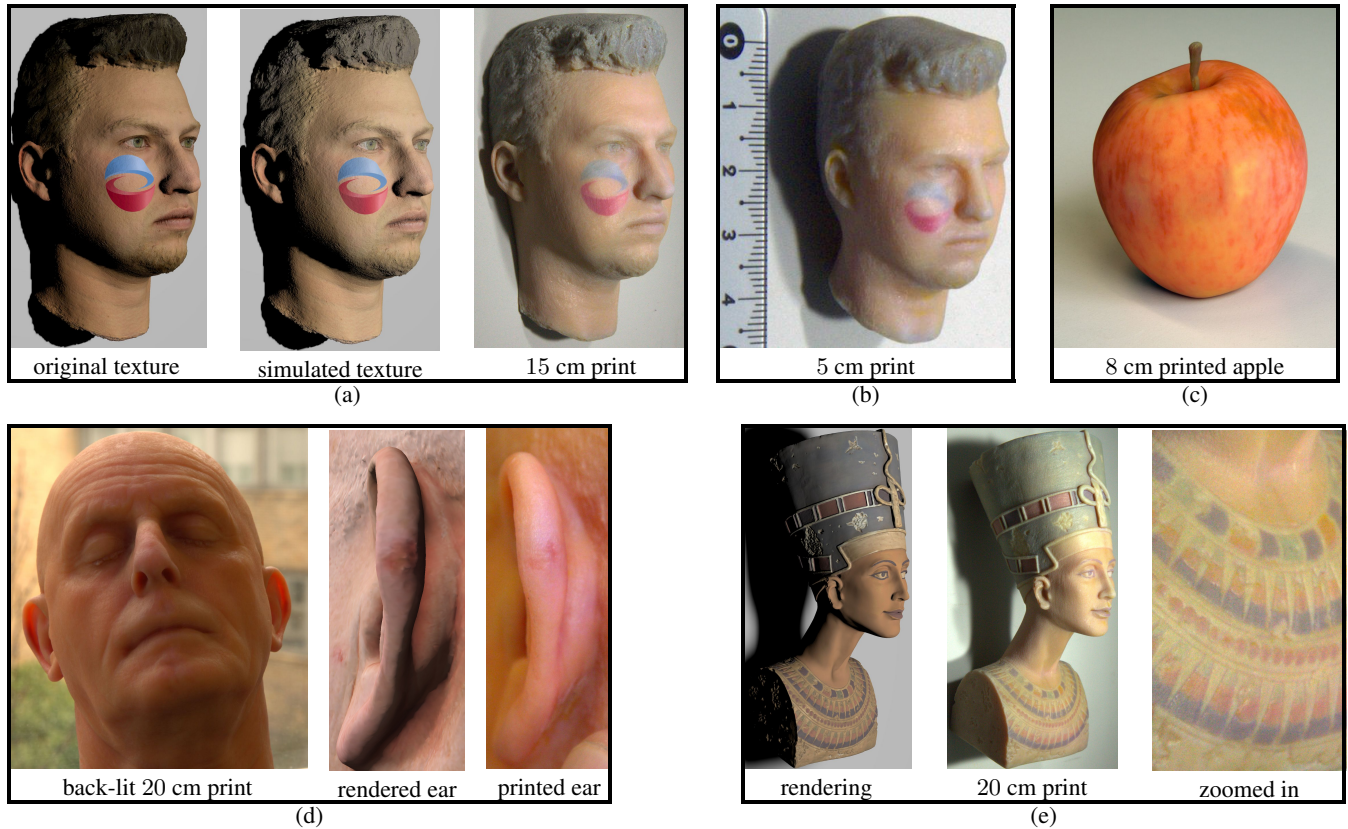


Fig. 12. Various 3D color prints generated using our software. Note that color deviations between rendering and print may have various reasons such as gamut limitations, goniochromatism and color inconstancy of printing materials, light field and illuminant differences between real scene and rendering, etc. The comparison aims to show how well texture details are preserved by our halftoning approach and that no artifacts are introduced. See the text for discussion of the individual parts.

Appendix

Our halftoning algorithms we propose are based on error diffusion, in which after a pixel or voxel is quantized, the resulting error is distributed to nearby pixels or voxels using an error diffusion filter. Thus, if a pixel with a value of 0.5 is quantized to 0, the error is distributed to its neighbors, making it more likely that they are quantized to 1. Formally, we are given an input tonal signal \mathbf{g} and an error signal \mathbf{a} , which is initialized $\mathbf{a}(\mathbf{v}) = \mathbf{0} \forall \mathbf{v}$. The error diffused signal is defined as

$$\tilde{\mathbf{g}}(\mathbf{v}) = \mathbf{g}(\mathbf{v}) + \mathbf{a}(\mathbf{v}), \quad (10)$$

which is then quantized using a threshold

$$\mathbf{h}(\mathbf{v}) = \begin{cases} 1 & \text{if } \tilde{\mathbf{g}}(\mathbf{v}) > t \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

the resulting error is then diffused to the neighbors

$$\mathbf{a}(\mathbf{u}) = \mathbf{a}(\mathbf{u}) - w_{\mathbf{v},\mathbf{u}} (\mathbf{h}(\mathbf{v}) - \tilde{\mathbf{g}}(\mathbf{v})) \quad (12)$$

where $\mathbf{u} \in \mathcal{N}(\mathbf{v})$ is a neighbor of \mathbf{v} , $w_{\mathbf{v},\mathbf{u}}$ is an element of the error diffusion filter diffusing error from \mathbf{v} to \mathbf{u} .

REFERENCES

- 3DSYSTEMS. 2014. Projet 860Pro. <http://www.3dsystems.com/3d-printers/professional/projet-860pro>.
- AGAR, A. AND ALLEBACH, J. 2005. Model-based color halftoning using direct binary search. *IEEE Trans. on Image Proc.* 14, 12, 1945–1959.
- ALEXA, M. AND KYPRIANIDIS, J. 2015. Error diffusion on meshes. *Computers and Graphics (Proc. SMI 2014)* 46, 336–344.
- ARIKAN, C., BRUNTON, A., TANKSALE, T., AND URBAN, P. 2015. Color-Managed 3D-Printing with highly Translucent Printing Materials. In *SPIE/IS&T Electronic Imaging Conference (accepted – included in supplemental material)*. San Francisco.
- BAYER, B. E. 1973. An optimum method for two-level rendition of continuous-tone pictures. In *IEEE Intl. Conf. on Comm.* Seattle, WA, 11–15.
- CAMPBELL, F., KULIKOWSKI, J., AND LEVINSON, J. 1966. The effect of orientation on the visual resolution of gratings. *The Jour. of Physiology* 187, 2, 427–436.
- CHANDRASEKHAR, S. 1960. *Radiative transfer*. Courier Dover Publications.
- CHANG, J., ALAIN, B., AND OSTROMOUKHOV, V. 2009. Structure-aware error diffusion. *ACM TOG (Proc. SIGGRAPH Asia)* 28, 5, 162:1–162:8.
- CHANG, T. AND ALLEBACH, J. 2003. Memory efficient error diffusion. *IEEE Trans. on Image Proc.* 12, 11, 1352–1366.

- CHEN, D., LEVIN, D., DIDYK, P., SITTHI-ARMORN, P., AND MATUSIK, W. 2013. Spec2fab: A reducer-tuner model for translating specifications to 3D prints. *ACM TOG (Proc. SIGGRAPH)* 32, 4.
- CHO, W., SACHS, E., PATRIKALAKIS, N. M., AND TROXEL, D. E. 2003. A dithering algorithm for local composition control with three-dimensional printing. *CAD* 35, 9, 851–867.
- DEMICHEL, E. 1924. Le procédé. 26, 3, 17–21, 26–27.
- DONG, Y., WANG, J., PELLACINI, F., TONG, X., AND GUO, B. 2010. Fabricating spatially-varying subsurface scattering. *ACM TOG (Proc. SIGGRAPH)* 29, 4.
- DOUBROVSKI, E., TSAI, E., DIKOVSKY, D., GERAEDTS, J., HERR, H., AND OXMAN, N. 2014. Voxel-based fabrication through material property mapping: A design method for bitmap printing. *CAD*.
- ESCHBACH, R. AND KNOX, K. 1991. Error-diffusion algorithm with edge enhancement. *JOSA A* 8, 12, 1844–1850.
- FAIRCHILD, M. D. 2013. *Color Appearance Models*, 3 ed. John Wiley & Sons, inc., West Sussex, England.
- FELZENZWALB, P. AND HUTTENLOCHER, D. 2004. Distance transforms of sampled functions. Tech. rep., Cornell University.
- FLOYD, R. AND STEINBERG, L. 1976. An adaptive algorithm for spatial grey scale. *Proc. of the Soc. of Info. Display* 17, 1.
- HAAŠAN, M., FUCHS, M., MATUSIK, W., PFISTER, H., AND RUSINKIEWICZ, S. 2010. Physical reproduction of materials with specified subsurface scattering. *ACM TOG (Proc. SIGGRAPH)* 29, 3.
- HERGEL, J. AND LEFEBVRE, S. 2014. Clean color: Improving multi-filament 3D prints. *CGF (Proc. Eurographics)* 33, 2, 469–478.
- ICC. 2010. *File Format for Color Profiles*, 4.3.0.0 ed. <http://www.color.org>.
- JOHNSON, G. AND FAIRCHILD, M. 2003. A top down description of S-CIELAB and CIEDE2000. *Color Research and Application* 28, 6, 425–435.
- LAN, Y., DONG, Y., PELLACINI, F., AND TONG, X. 2013. Bi-scale appearance fabrication. *ACM TOG (Proc. SIGGRAPH)* 32, 4.
- LAU, D., ARCE, G., AND GALLAGHER, N. 1999. Digital halftoning by means of green-noise masks. *JOSA A* 16, 7, 1575–1586.
- LAU, D. L. AND ARCE, G. R. 2001. *Modern digital halftoning*. CRC Press.
- LEVIN, A., GLASNER, D., XIONG, Y., DURAND, F., FREEMAN, W., MATUSIK, W., AND ZICKLER, T. 2013. Fabricating BRDFs at high spatial resolution using wave optics. *ACM TOG (Proc. SIGGRAPH)* 32, 4.
- LI, P. AND ALLEBACH, J. 2004. Tone-dependent error diffusion. *IEEE Trans. on Image Proc.* 13, 2, 201–215.
- LOU, Q. AND STUCKI, P. 1998. Fundamentals of 3D halftoning. *LNCS (Proc. Elect. Pub. and Art. Imag.)* 1375, 224–239.
- MCCOR TECHNOLOGIES. 2014. MCor Iris. <http://mccor technologies.com/3d-printers/iris/>.
- MITSA, T. AND PARKER, K. 1992. Digital halftoning technique using a blue-noise mask. *JOSA A* 9, 11, 1920–1929.
- MOROVIČ, J. 2008. *Color Gamut Mapping*. John Wiley & Sons.
- MULLEN, K. T. 1985. The contrast sensitivity of human colour vision to red-green and blue-yellow chromatic gratings. *The Jour. of Physiology* 359, 1, 381.
- OSTROMOUKHOV, V. 2001. A simple and efficient error-diffusion algorithm. In *Proc. SIGGRAPH*.
- PANG, W.-M., QU, Y., WONG, T.-T., COHEN-OR, D., AND HENG, P. 2008. Structure-aware halftoning. *ACM TOG (Proc. SIGGRAPH)* 27, 3, 89.
- REINER, T., CARR, N., MECH, R., STAVA, O., DACHSBACHER, C., AND MILLER, G. 2014. Dual-color mixing for fused deposition modeling printers. *CGF (Proc. Eurographics)* 33, 2, 479–486.
- ROGERS, G. L. 1997. Optical Dot Gain in a Halftone Print. *JIST* 41, 643–656.
- SCHMIDT, R., GRIMM, C., AND WYVILL, B. 2006. Interactive decal compositing with discrete exponential maps. *ACM TOG (Proc. SIGGRAPH)* 25, 3, 605–613.
- STRATASYS. 2014. Objet500 Connex3. <http://www.stratasys.com/3d-printers/production-series/connex3-systems>.
- SÜSSTRUNK, S., BUCKLEY, R., AND SWEN, S. 1999. Standard RGB color spaces. In *IS&T/SID, 7th CIC*. Scottsdale Ariz., 127–134.
- ULICHNEY, R. 1987. *Digital Halftoning*. The MIT Press.
- VAN NES, F. L. AND BOUMAN, M. A. 1967. Spatial modulation transfer in the human eye. *JOSA* 57, 3, 401–406.
- VIDIMČE, K., WANG, S.-P., RAGAN-KELLEY, J., AND MATUSIK, W. 2013. Openfab: A programmable pipeline for multi-material fabrication. *ACM TOG (Proc. SIGGRAPH)* 32, 4.
- VORA, P. L. AND TRUSSELL, H. J. 1993. Measure of goodness of a set of color-scanning filters. *JOSA A* 10, 1499–1508.
- WYBLE, D. R. AND BERNS, R. S. 2000. A Critical Review of Spectral Models Applied to Binary Color Printing. *Color Research and Application* 25, 1, 4–19.
- WYSZECKI, G. AND STILES, W. 2000. *Color Science: Concepts and Methods, Quantitative Data and Formulae*, 2 ed. John Wiley & Sons, inc.
- ZHOU, B. AND FANG, X. 2003. Improving mid-tone quality of variable-coefficient error diffusion using threshold modulation. *ACM TOG (Proc. SIGGRAPH)* 22, 3, 437–444.

9. SUPPLEMENTAL MATERIAL

9.1 Radiative transfer

Given an arrangement of multiple non-fluorescent printing materials with similar refractive indexes within a shape \mathcal{S} , light propagation within this shape can be described by the steady-state radiative transfer equation [Chandrasekhar 1960]

$$\frac{dI_\lambda(x, \Omega)}{dx} = -a_\lambda(x)I_\lambda(x, \Omega) - s_\lambda(x) \left(I_\lambda(x, \Omega) - \oint_{\mathbb{S}^2} I_\lambda(x, \Omega') p_\lambda(\Omega', \Omega) d\Omega' \right) \quad (13)$$

where $I_\lambda(x, \Omega)$ is the radiant intensity at location $x \in \mathcal{S}$ propagating in direction Ω for wavelength $\lambda \in [380, 730]$ nm (visible wavelength range), $a_\lambda(x) \geq 0$ is the spectral absorption coefficient, $s_\lambda(x) \geq 0$ is the spectral scattering coefficient, \mathbb{S}^2 is the sphere and $p_\lambda(\Omega', \Omega)$ is the scattering function satisfying $\oint_{\mathbb{S}^2} \oint_{\mathbb{S}^2} p_\lambda(\Omega', \Omega) d\Omega' d\Omega = 1$. Solving the radiative transfer equation with an appropriate boundary condition at the surface gives us the radiation that is diffusely emitted from the shape. Adding the radiation directly reflected from the surface due to Fresnel reflection, gives us the total radiation emitted from the object's surface.

Eq. (13) shows that the intensity of radiant energy traveling through the material is attenuated by absorption (note that $a_\lambda(x)$ depends on the wavelength but is independent of the traveling direction) and is redistributed by scattering (scaled by $s_\lambda(x)$). Thus, a fraction of light entering the print at one location may be emitted from the surface at a different location due to scattering. If light travels through different materials its spectral power distribution is modulated by each material's absorption coefficients and the path length within this material.

9.2 Gamut Volume Visualizations

Figure 13 shows the gamut volumes from the paper projected onto the a^*b^* plane to provide another perspective on how the gamut volume changes as layers are added.

9.3 Visual Experiment Setup

Figure 14 shows the physical setup we used to conduct the visual experiment. Subject placed their chin on the chinrest and viewed the printed object next to a rendering of it on a color-calibrated display. The color characterization of the display was performed from the chinrest position considering the viewing booth illumination. In this way, tristimulus values from objects placed on the display and illuminated by the viewing booth could be reproduced on the screen if viewed from the chinrest position.

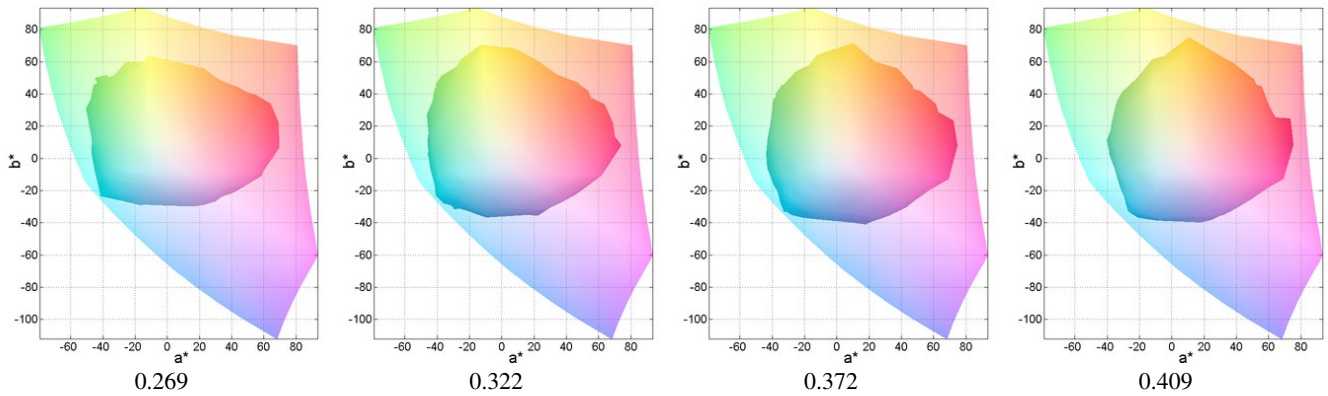


Fig. 13. Increasing number of layers results in increasing gamut volume. From left to right: 3, 6, 12 and 18 layers. The gamut volume is indicated below each image as a fraction of the volume of sRGB gamut shown transparent.

Experimental Setup

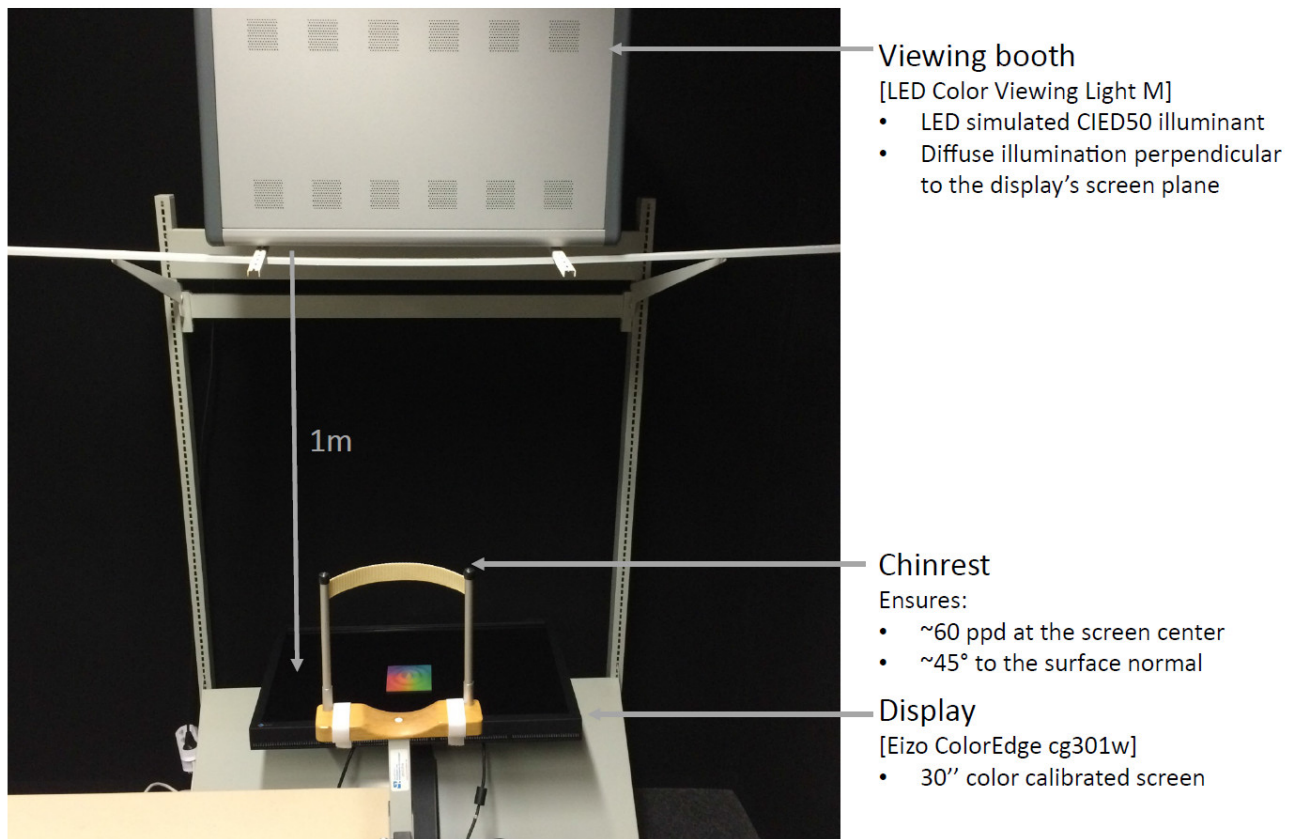


Fig. 14. Physical setup for our visual experiment.