Computers & Graphics 36 (2012) 635–641

Contents lists available at SciVerse ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Section on CANS

Irregular pit placement for dithering images by self-occlusion

Marc Alexa [a,c,*], Wojciech Matusik [b,c]

[a] TU Berlin , Germany & University of Toronto, Canada
[b] MIT, Boston, MA, USA
[c] Disney Research Boston ,Boston, MA, USA

ARTICLE INFO

Article history:
Received 14 November 2011
Received in revised form
13 February 2012
Accepted 14 February 2012
Available online 3 March 2012

Keywords:
Dithering
Stippling
Scanning
3d output

ABSTRACT

We propose a complete system for designing, simulating, and fabricating surfaces with shading due to self-occlusion that induce desired input images. Our work is based on a simple observation. Consider a cylindrical hole (a pit) in a planar surface. As the depth of the hole increases, the radiance emitted from the surface patch that contains the hole decreases. This is because more light is trapped and absorbed in the hole. First, we propose a measurement-based approach that derives a mapping between average albedo of the surface patch containing the hole and the hole depth. Given this mapping and an input image, we show how to produce a distribution of holes with varied depth that approximates the image well. We demonstrate that by aligning holes with image features, while still maintaining an irregular distribution, we can obtain reproductions that look better than those resulting from regular hole patterns—despite using less holes. We validate this method on a variety of images and corresponding surfaces fabricated with a computer-controlled milling machine and a 3D printer.

1. Introduction

Artists have long been experimenting with the visual effect of self-occlusion on surfaces. For example, in sculpting, chisel marks on surfaces cause shading due to occlusion. A skilled artist can use this shading to induce an apparent desired texture even on surfaces made of a uniform material. In our work we explore a very similar concept. Our goal is to automatically produce a model of a surface made of a uniform material that can reproduce a desired input texture or an image. Once we generate a model of this surface, we can use automated devices such as computer-controlled CNC milling machines or 3D printers to physically fabricate the real surface. This paper shows a complete pipeline (and a suite of tools) that allows converting an arbitrary image (e.g., a photo taken with a digital camera) to a corresponding surface that can be physically fabricated from a range of different substrates (e.g., ceramic, plaster, wood, or metal).

Apart from its potential aesthetic appeal, this process has interesting practical applications as many consumer products are mass-manufactured using an injection molding method. In this method a substrate is injected into a custom pre-manufactured mold. By minimally modifying the mold, our method allows the addition of grayscale images to virtually any injection-molded product at almost no additional cost. In comparison, adding graphics or images to these surfaces requires an expensive, additional painting procedure that is often performed manually.

Our process uses cylindrical pits as basic geometric primitives (although this method can be easily extended to accommodate different primitives as well). We make a simple observation: when we increase the depth of a pit, the average apparent albedo of the surface patch containing this pit decreases. This is because a deep pit generally absorbs more light than a shallow one. One of our goals is, clearly, to optimize the apparent match between the computer prediction (or simulation) and the appearance of the real surface. For predicting the dependence between hole depth and albedo we use a simple data-driven solution. We first manufacture a calibration pattern with holes of different depths. Then, we use a measurement system to estimate an apparent albedo as a function of hole depth. Next, we build an inverse function that maps apparent albedos to the hole depth. This inverse function allows us to go directly from a desired image to a pitted surface based on a regular grid and thus a desired surface model.

The main point of this paper, however, is to investigate irregular hole patterns. Irregular grids allow us to improve the resolution mismatch between the number of pits in the fabricated surface and the number of pixels in the input image by aligning pits with the dominant image features. We describe a new variant of Centroidal Voronoi Tessellations (CVT) with identical areas for the pit centers—an approach so far suggested by Balzer et al. [1] for dithering and stippling. Our version, however, uses a different algorithm and also computes centroids in weighted fashion, to

* Corresponding author at: TU Berlin , Germany & University of Toronto, Canada.
E-mail address: marc.alexa@tu-berlin.de (M. Alexa).

0097-8493/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2012.02.008

Fig. 1. Input images, a pattern of pits and the corresponding power cells.

avoid bad locations for the pit centers. Since regular grids provide the maximum number of holes per area this process necessarily decreases the number of holes. Yet, the results from irregular patterns are clearly better than those based on regular grids.

We believe that our approach has several interesting and unique properties. First, the process is completely ink-free—we can embed arbitrary images into almost any surface and without using any dye or additional material. The approach is relatively inexpensive, simple to implement, and works well on large-scale surfaces. The produced surfaces are durable and integrate well with current materials. We also believe that these surfaces have a very unique feel and they can be appreciated by many observers. Finally, this method could also be used in conjunction with surfaces with non-uniform albedo (e.g., to increase the dynamic range of the medium).

2. Related work

The problem of computing a surface that depicts a given image based on the occlusion of small holes is original. To the best of our knowledge, the problem has not been explicitly stated and thus there are no methods that have ever attempted to solve it. However, there are a few related publications in the computer graphics and computer vision literature.

In particular, the problem of predicting an aggregate surface reflectance of diffuse surfaces with various micro-geometries has been explored. For example, Oren and Nayar [2] develop a model for Lambertian microfacets. Koenderink et al. [3] predict an aggregate reflectance of surfaces with spherical concavities. Similarly Mérillou et al. [4] develop a reflectance model with holes inserted into the surface profile. In this work, we do not deal at all with angular reflectance variation, instead we focus on spatial reflectance properties. Computer graphics researchers have also investigated the problem of generating bas-reliefs [5–8]—an automated process of embedding three-dimensional models in the form of reliefs onto planar surfaces. However, the visual result of the relief depends on the fact that compression of depth values is tolerated by the human visual system (the bas-relief ambiguity [9]). In a recent approach, we have created reliefs for specific light directions, yet not based on shading [10].

There have been also a number of methods that attempt to reproduce images on uniform materials. For example, one method [11,12] computes a spatially varying thickness of a uniform material slab such that the desired image is induced by the material transmission. Perhaps the most similar work to ours is the shadow art by Mitra and Pauly [13]. They generate a volumetric structure that casts three different user-specified binary shadows onto planar surfaces. In our case, we generate a planar structure that would induce a single grayscale image directly on its surface using self-occlusion.

For generating irregular pit distributions, we draw from algorithms for computer-generated stippling patterns [14–17]. In particular capacity-constrained diagrams [1,18] have inspired the our current algorithm, as it also considers equal area of cells an important goal. However, there are some important differences compared to our setting. First, in our method as we are also able to change the grayscale level of each stipple continuously while the typical stipples have no intensity variation, or they are quantized [19]. Second, we have to absolutely enforce the distance between the stipples in order to be able to manufacture the physical surface without losing the property of self-occlusion.

3. Model

As a first step for creating and analyzing surfaces we need to define what we understand as the image generated by the pitted surface. We assume all pits to have the same radius r and further that no two pits intersect. Let $P=\{(\mathbf{p}_c,p_v)\}$ represent the pits, with $\mathbf{p}_c$ being the position of the center of the pit p and p_v being the associated value depending on the depth of the pit and the associated area.

We associate the power cell [20] with the radius r located at p_c to the pit and model the gray value in each cell as constant. A power cell is defined as the set of points closest to p_c based on the pseudo-distance function

$$d_p(\mathbf{x}) = \|\mathbf{p}_c - \mathbf{x}\|^2 - r^2. \tag{1}$$

It turns out the shape of the power cells remains unchanged by adding a constant to all radii r. This means that the diagram we need to compute is equivalent to the Voronoi diagram,[1] as long as the radii are constant.

Fig. 1 shows how Voronoi regions are fitted to the pits. This allows us to compute the necessary gray values, and thus depths, for the pits.

Based on this model, we can (independently) consider the problems of mapping the tone values in the image to the physical artifact and placing the individual pits on the surface.

4. Computing values and depth

First, we describe our data-driven model generating a table that converts from the values p_v to the necessary depth of the hole. The table depends on the material and we describe a simple measurement process to estimate it. It turns out that the dynamic range is quite limited. Consequently, we also explain how to map an image into this dynamic range.

[1] We note that using the power diagram has several conceptual advantages over considering the Voronoi diagram of the circular pits, as we did before [21].

4.1. Data-driven pit model

We would like to produce the best possible match between the desired image and the radiance values observed by the viewer. However, these values depend on the viewer's position and the illumination environment. In our setup, we assume the viewer is observing the surface directly from the front (perpendicular to the surface). It turns out that the exact position of the viewer is not critical in practice (see Fig. 6). It is, in general, difficult to know the typical illumination conditions beforehand (e.g., the surface can be viewed under different lighting conditions). Therefore, we assume that the incident lighting is uniform. One could also optimize the surface for a particular incident lighting. In this case this incident lighting would have to be recreated in the measurement setup.

In our procedure, we have to manufacture a calibration pattern for each material type. We use this pattern to determine the dependence of the apparent albedo of a surface patch on the pit depth. Our pattern consists of several rows of equal lines of pits in ascending depths from zero to a maximum depth. We use a white diffuse tent to approximate the uniform illumination. The camera is placed directly above the surface. We also use a material sample with known reference albedo and we estimate the albedo of our surface by comparing it to the reference material.

Next, we process the calibration image in order to compute the mapping between the apparent albedo and the pit depth. In particular, we use a hexagonal box filter that covers a Voronoi region of each pit. Integrating over this region yields the average value or the effective albedo of the pit and its surrounding area. We obtain the absolute value by comparing this value to the calibration white albedo target captured in the same image. We show a sample calibration target and the corresponding processed image with applied filter in Fig. 2. Once the average apparent albedos for each pit depth are measured it is straightforward to convert this function to an inverse mapping. As we might not have the measurements for all necessary pit depths, we approximate the intermediate values with linear interpolation (see Fig. 2). We also ensure that the function is monotonically decreasing with the increasing pit depth. Thus, we obtain a function that relates an average apparent albedo to the pit depth.

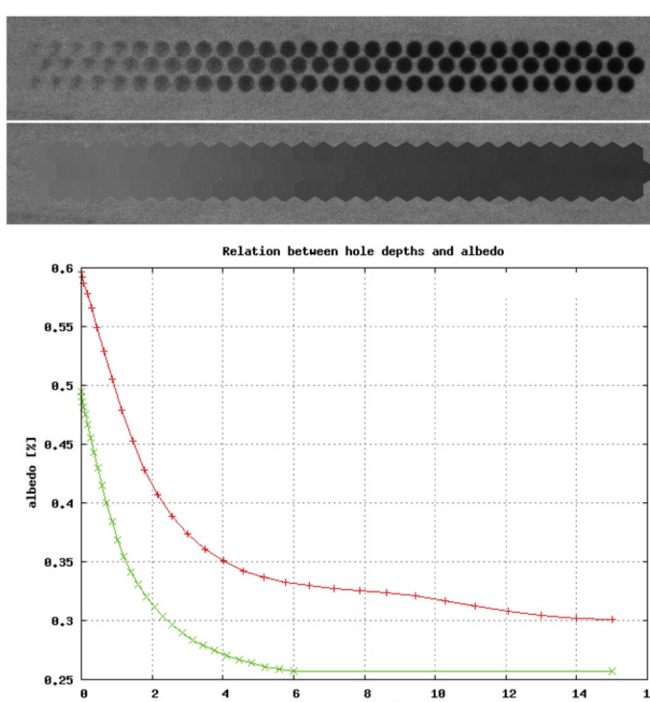

Fig. 2. A drilled calibration pattern in a wood sample and the fitted cells (above). A measurement-based estimate of average apparent albedo for wood (green) and a 3D printing polymer (red) as a function of pit depth (below). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.2. Tone mapping

Once we have computed the inverse mapping for a given material, we can easily simulate what a surface corresponding to a given image is going to look like without even fabricating a real surface. This is extremely useful as it is much easier and faster to run this simulation.

However, before we simulate the appearance of the surface we need to tone-map the desired input image to the available range of a given substrate material. For example, in the case of wood the available dynamic range is from 0.25 to 0.50. Thus, the range is limited both on the bright and the dark end. One simple method to deal with this limited range is to remap input images (their minimum and maximum used value) to the whole usable range of the material. However, it is also useful to employ more advanced tone-mapping operators for this task. In particular, we use the operator proposed by Mantiuk et al. [22].

5. Pit distribution

The densest packing for circular elements in the plane is the hexagonal grid. However, it turns out because of the very small resolution of the pitted image, it is almost always better to place the pits in an irregular fashion. We explain the rational of our placement process and how it significantly improves over an earlier version presented in the conference version of this work [21].

5.1. Desirables

There are two important considerations for the placement of the pits:

1. Image error, i.e. the distance between the image model for the pitted surface and the true image.
2. Distribution regularity, i.e. a distraction from the image content by apparent regularities or irregularities in the pit distribution.

In an earlier algorithm [21] we had successfully reduced the image error relative the hexagonal grid by placing the pit centers away from sharp corners in the image. While this was the rationale of the algorithm, it simply minimized the image error in a discrete and iterative fashion. This lead to regular pit placement inside homogeneous image regions and irregularities around image discontinuities. From the perspective of minimizing the image error this makes perfect sense. However, many observers told us they found the apparent differences in modeling different regions in the image quite distracting. This effect can be similarly found in dithering, where the goal is always to avoid any discernible patterns. We provide a new algorithm for our problem here. This algorithm tries to balance between the desired properties mentioned above.

5.2. Basic distribution strategy

Given a surface area, and the desired number of pits to placed, we wish to place the pits such that their image model is close to the desired image and the placement is irregular. It has been shown by Balzer et al. [1] that Voronoi diagrams with equal area sites are well suited for the task of avoiding regularities while still providing a good distribution. They model the requirement of equal area in a discrete way (i.e. by assigning the same number of discrete elements to each site) and then minimize other energy functions.

In general, however, the equal area constraint can be modeled also in a continuous fashion [23]. We use power cells and make the following observation: by making the radius of a cell larger, its area grows (and vice versa). Exploiting the mechanism allows one to essentially prescribe the areas of the cells for any point distribution (possibly at the expense of disconnecting cells).

We generally follow the idea of computing a centroidal distribution with equal areas. However, our algorithm does this optimization in a continuous fashion, by updating the position and radii of the pits concurrently. We use a different power weight[2] for each pit. Let w_p be the power weight for cell p, then we measure the distance to a point $\mathbf{x}$ from the cell center $\mathbf{p}_c$ as

$$d_p(x) = \|\mathbf{p}_c - \mathbf{x}\|^2 - w_p^2. \tag{2}$$

Our goal is to assign the same area to each pit cell, while moving each pit center into the centroid of its cell. This is known to generate well balanced distributions. A result of our algorithm for this strategy can be found in the first row of Fig. 5.

5.3. Improving pit center placement

Of course, our assumption is that each cell has a constant color. This means that the image reproduction depends on the placement of the pit centers. If all pit centers are placed on a piece of constant image function, the reproduction is good. If a pit center lies on a sharp corner, this corner is lost because of the constant color in the corresponding cell. Consequently, it is desirable to locate pit centers away from sharp image corners (as often done in other image processing applications, e.g. mosaic edges [24]). More precisely, we need to model the error associated with different locations for the pit centers and find an (almost) equal area distribution of the pits such that the pit centers are located away from high error locations.

One could compute the error associated with a cell by considering the squared sum of differences of associated image values. We approximate this function a priori by considering the constant radius r. For each pixel, we compute the desired average tone value from all pixels within r. For this we first simply count the number of pixel in radius r around (x,y) and denote this number as $n_r\,(x,y)$. Then the average image value is

$$A(x,y) = \frac{\sum_{(s-x)^2+(t-y)^2 \le r^2} I(s,t)}{n_r(x,y)}, \tag{3}$$

and this leads to the error as

$$E(x,y) = \left(\frac{\sum_{(s-x)^2+(t-y)^2 \le r^2} (A(x,y)-I(s,t))^2}{n_r(x,y)} \right)^{1/2}. \tag{4}$$

This discrete map is computed at the beginning of the computation and then stored. Fig. 4 shows the E for a few images used in the analysis section. We extend the discrete images to the continuous domain for value pairs (x,y) by bilinear interpolation.

We want to get a good pit distribution while trying to minimize $E(x,y)$ for all pit centers. We do this in the following Lloyd-inspired algorithm: in each cell, compute a *weighted* centroid, and the weights aim at avoiding high-error locations. We have chosen to compute the centroid as an approximation to $\iint \exp(-\lambda E^2(x,y))\, dx\, dy$, where the coordinates are weighted by the error function $E(x,y)$. The parameter λ is user chosen. Setting λ to zero allows computing an image independent distribution. As λ gets larger, pit centers are more and more located away from areas with large error.

For the approximation of the integral, we first compute the centroid of the vertices of the cell. We use the location of the centroid to decompose the cell into triangles. For each triangle, again, the centroid is computed. These triangle-centroids are weighted by triangle area and the terms $\exp(-\lambda E(x,y))$, where (x,y) is the location of the centroid.

5.4. Lloyd-type algorithm

Lloyd's algorithm [25] computes centroidal Voronoi distributions by repeatedly moving each center to the centroid of its cell. Our algorithm does the same in directions of x and y, except for computing the centroid as explained above (see the details in the pseudo-code further below).

The radius $w_p \in \mathbb{R}$ of each cell is updated in a similar fashion to move towards an equal-area distribution. We make the update linear in the difference between desired area of the cell and the actual area, i.e. quadratic in the difference relative to the *weight* of the power cell: $w_p = w_p \pm dA_p^2$, where the sign depends on the sign of dA_p. We have found that larger movements are hard to control and often hinder convergence.

The following pseudo-code summarizes the move of each pit center. We note that we have implemented the power diagram as a three-dimensional Voronoi diagram, intersecting the cells against the $x-y$ plane. For the clarity of the pseudo-code, we are not exposing this implementation detail here and rather describe the optimization in terms of a power diagram.

MOVE (*pit cell p, parameter* λ)

Translate the centroid of vertex coordinates to $(0,0)$
$\mathbf{p}_c = A = W = 0$
for each vertex coordinate (x_i, y_i) in the power cell of p
  $a_i = \frac{1}{2}(x_i y_{i+1} - x_{i+1} y_i)$
  $\mathbf{c}_i = \frac{1}{3}((x_i, y_i) + (x_{i+1}, y_{i+1}))$
  $\omega_i = exp(-\lambda E(c_i))$
  $\mathbf{p}_c = \mathbf{p}_c + \omega_i a_i \mathbf{c}(i)$
  $A_p = A_p + a_i$
  $\Omega = \Omega + \omega_i a_i$
$\mathbf{p}_c = \mathbf{p}_c / \Omega$
$dA_p = 1/(\text{image area}) - A_p$
if $(dA_p > 0)$
  $w_p = w_p - dA_p^2$
else
  $w_p = w_p + dA_p^2$

Note that it would always be possible to make all areas equal while holding the positions in the (x,y) plane constant as a post process. However, since the pit centers are not affected, it is not necessary to do so. We have found that the algorithm converges in areas and positions for the standard weighting $(\lambda = 0)$, while the areas are not converging to an equal size for weights larger zero. Nevertheless, we find the distributions to be very useful for our purpose.

This algorithm moves very quickly towards distributions that are useful in practice. Our unoptimized implementation runs on the order of 100 s for the 10,000 points on a standard laptop computer. This is significantly faster than the older version of this algorithm, which was on the order of several minutes to hours [21], and is negligible compared to the time for generating the physical artifact.

[2] This weight is also called radius in the literature, however, we avoid this term because of the potential confusion with the pit radius.

6. Results

In this section, we first describe how to simulate the surface without actually fabricating it. This helps for comparing and analyzing the distribution of pits. Next we discuss the surface fabrication process. Then we present the examples of fabricated surfaces. Finally we discuss several limitations of the method.

6.1. Surface prediction

Our pitted surfaces either use hexagonal grids or the optimized placement just described. For a first approximation of the result we created a plug-in for the PBRT raytracer from Pharr and Humphrey [26]. While this allows simulating the pit effects fairly well it is relatively difficult to capture the exact characteristics of the material we used and other artifacts from the drilling process within the raytracer framework. Furthermore, simulating pits correctly with proper shading, internal reflections inside the pits, and subsurface scattering is extremely expensive computationally and more measurements of the surface properties would be necessary.

Therefore, we resort to an image-based rendering approach. This approach proves to be extremely simple but it predicts the appearance of the output images very well. Thus, we compute a mapping, where for each apparent albedo value, we also store a small sprite corresponding to the surface patch including a pit and its surrounding area. Thus, in order to assemble the whole simulated image we simply composite sprites corresponding to the appropriate albedo values.

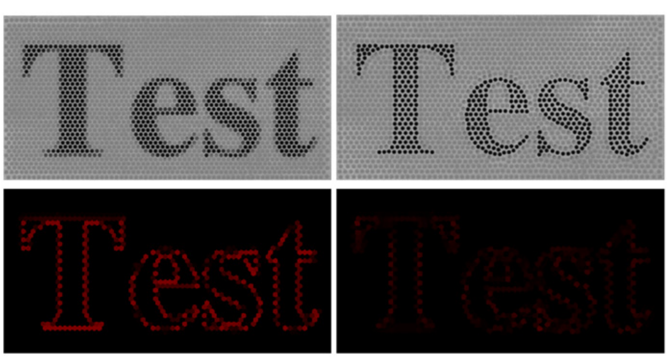

Fig. 3. Comparison of the uniform hexagonal tessellation (left) and the irregular placement (right) from our previous algorithm. While the error towards the input image is reduced (bottom row) note the regular and irregular pattern in the result.

6.2. Comparison of pit distributions

The final self-occlusion surface is described as a set of pit centers $\mathbf{p}_c$ computed as explained in Section 5. We have compared the regular (hexagonal) pit distribution to the irregular distribution on several examples based on our earlier approach. It is obvious that the piecewise constant reproduction severely suffers from undersampling (e.g. see also Fig. 3).

In the following we compare several pit distributions for the algorithm described before with varying λ.

6.3. Fabrication of surfaces

We have experimented with fabricating the surfaces using two different methods: by drilling holes using a CNC machine and by using a 3D printer. In order to manufacture pitted surfaces by drilling we have used a MicroMill 2000 by MicroProto Systems. This is an inexpensive desktop Computer Numerical Control milling machine with three axes of motion. The maximum surface size that can be manufactured is 225 mm $\times$ 146 mm. Moreover, the machine is repeatable up to 0.01 mm—this is certainly enough for our task. We use a desktop PC running Mach3 control software to execute GCode (RS274D) [27]—a standard command language for CNC machine tools. On our machine a typical image of about 10,000 pits resolution takes roughly 5–7 h depending on the hole depths. Our milled examples have sizes of about 130 mm $\times$ 130 mm. It would be straightforward to produce larger surfaces with more pits and even faster when using a commercial milling machine.

We experimented mostly with wood as it is inexpensive and moderately well suited for milling being not particularly strong but on the other hand not demanding on machine strength and cooling efficiency. Obviously, wood is a suboptimal choice for reproducing a good dynamic range but it has a nice look and feel as a physical material. We could imagine that other materials with higher albedo would allow significant improvements of the dynamic range.

We also experimented with rapid prototyping methods as 3D printing. For this task, we have used the OBJET Connex 350 3D printer. The maximum working range for this printer is 350 mm $\times$ 350 mm. Furthermore, the printer offers excellent precision (e.g., the resolution in x, y, and z direction is 30 μm). All our printed samples are more than 100 mm in length and width. We have used the holes with diameter of 1.2 mm and the thickness of the walls is 0.3 mm. Since the material used by this printer is quite translucent we have decided to air-brush all samples with a thin layer of uniform white diffuse paint. In Fig. 6 we show how this surface behaves under different viewing conditions.

One could also manufacture surfaces and objects using injection molding methods. These methods could deliver at least an

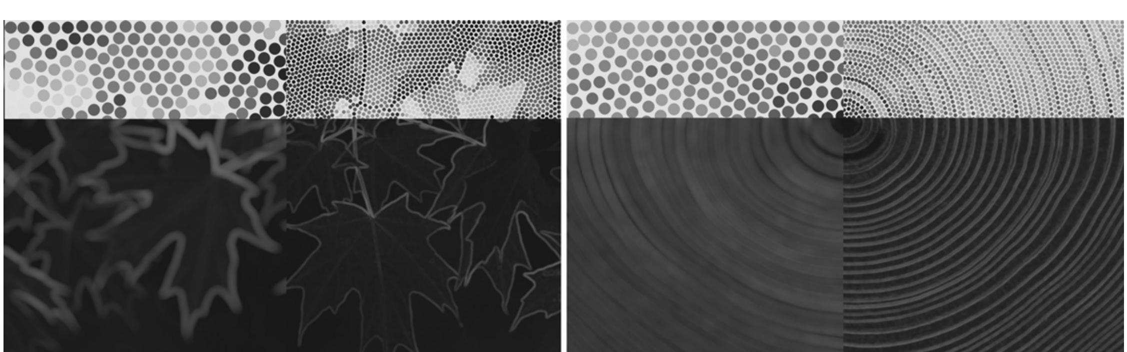

Fig. 4. The images show the function $E(x,y)$ for two different radii, illustrated by the pits on the top of the images.

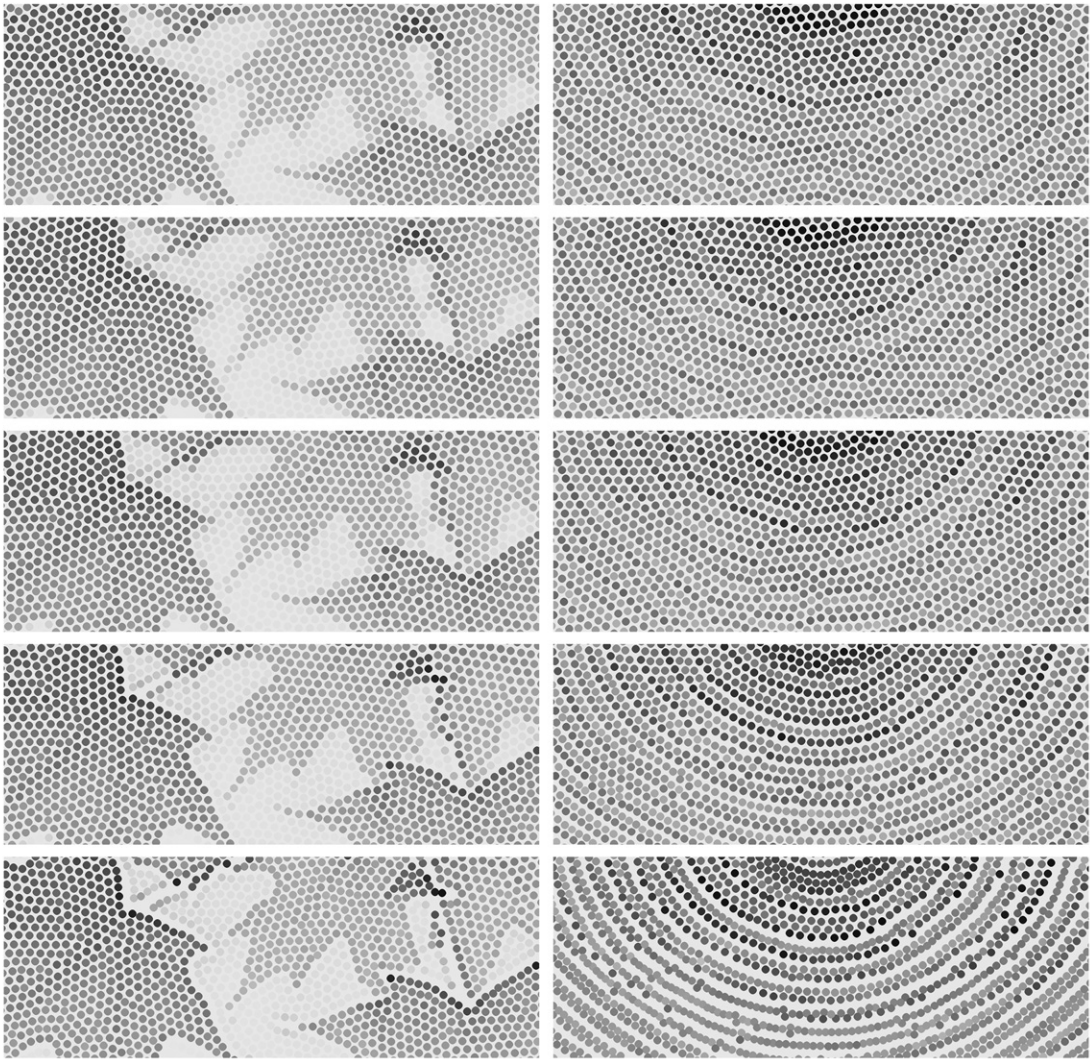

Fig. 5. Simulated results for 10,000 pits and $\lambda = \{0, 1, 2, 4, 16, 64\}$ (see Section 5.3 for details on the effect of the parameter).

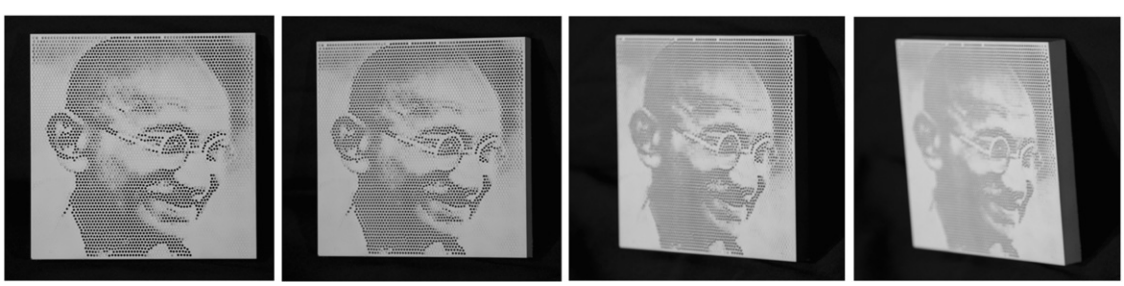

Fig. 6. The photographs of the surface depicting Ghandi fabricated using a 3D printer. The image is clearly visible from different viewpoints.

order of magnitude higher resolution compared to the methods we have used. The only downside is the high cost of manufacturing the molds, which is typically amortized by mass-producing many copies.

6.4. Limitations

Our method has a few limitations. One of the major limitations is the limited contrast. One can improve the working range by using materials with large albedo values. The average albedo value of the wood we use is around 50%. We can easily obtain surfaces with higher albedo (90% or higher) such as

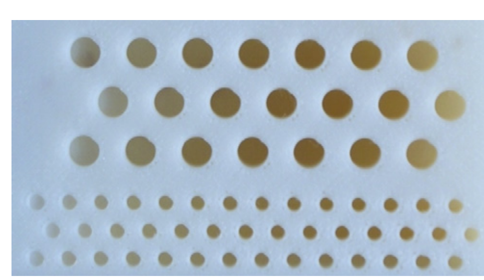

Fig. 7. One of major limitations of our method is subsurface scattering. Materials with large subsurface scattering will have poor contrast when using holes with small diameter.

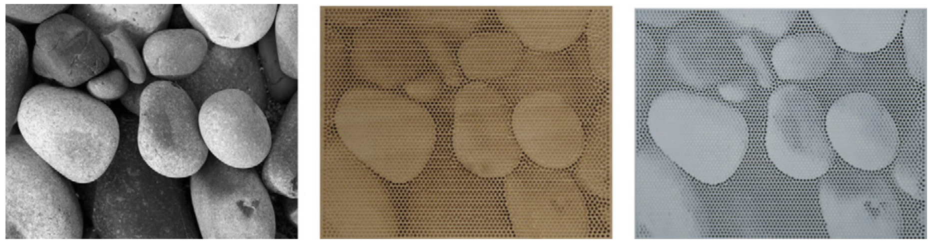

Fig. 8. An image of stone pebbles (left) and the results drilled in wood (center) and 3D printed using an optimized irregular drill pattern (right).

alumina oxide. The second way of improving contrast is to use holes with very thin walls—decreasing the relative surface area corresponding walls will improve the overall contrast. However, having surface with very thin walls might cause material damage.

Another important limitation is subsurface scattering (see Fig. 7). In the case of materials with large subsurface scattering, the contrast can decrease. This is because light is scattering inside of the material and can leak through the walls. The solution to this problem is using materials with low subsurface scattering or using large holes, for which subsurface scattering has smaller effect. Finally, the contrast of the pitted surface decreases when looking at the surface from oblique angles.

7. Conclusions and future work

This paper proposes a solution to embed arbitrary images in a material with a uniform albedo. We exploit shading due to self-occlusion to design, simulate, and fabricate surfaces that induce desired images. We use a data-driven measurement setup in order to correctly predict the effect of self-occlusion caused by cylindrical pits embedded in a surface. We explore both regular and irregular pit distribution on a surface in order to approximate a given image. We show a number of different simulated surfaces and also corresponding surfaces fabricated with a CNC milling machine and a 3D printer (see Fig. 8 and earlier examples). We believe that this method provides designers, artists, hobbyist with a powerful tool to create and fabricate custom surfaces. Moreover, the method can be used to add images to objects or surfaces that are mass-produced using injection molding.

We believe that there are many possible avenues for future work. First, we would like to explore pits with different diameter (as opposed to fixed diameter pits) and pits with more general cross-sections. This would allow us to increase the dynamic range (as the wall area is fixed, the ratio of the pit area to the total area is increasing with larger diameter). Pits with different cross-sections could improve image reproduction quality. Another direction is to modify arbitrary 3D models (instead of planar surfaces) in order to enrich them with arbitrary grayscale textures. Finally, we believe it is possible to add color to the process by using a multi-layer materials, where each layer has a different color.

References

[1] Balzer M, Schlömer T, Deussen O. Capacity-constrained point distributions: a variant of Lloyd's method. ACM Trans Graph 2009;28:86:1–8. doi:10.1145/1531326.1531392.

[2] Oren M, Nayar SK. Generalization of Lambert's reflectance model. In: Proceedings of the 21st annual conference on computer graphics and interactive techniques, SIGGRAPH '94, ACM, New York, NY, USA; 1994. p. 239–46, doi:http://10.1145/192161.192213.

[3] Koenderink JJ, Doorn AJV, Dana KJ, Nayar S. Bidirectional reflection distribution function of thoroughly pitted surfaces. Int J Comput Vis 1999;31(2–3):129–44.

[4] Mérillou S, Dischler J-M, Ghazanfarpour D. A BRDF postprocess to integrate porosity on rendered surfaces. IEEE Trans Visual Comput Graph 2000;6(4):306–18. doi:10.1109/2945.895876.

[5] Cignoni P, Montani C, Scopigno R. Computer-assisted generation of bas- and high-reliefs. J Graph Tools 1997;2:15–28.

[6] Weyrich T, Deng J, Barnes C, Rusinkiewicz S, Finkelstein A. Digital bas-relief from 3d scenes. ACM Trans Graph 2007;26(3):32:1–8. doi:10.1145/1276377.1276417.

[7] Song W, Belyaev A, Seidel H-P. Automatic generation of bas-reliefs from 3d shapes. In: Proceedings of the IEEE international conference on shape modeling and applications 2007, IEEE Computer Society, Washington, DC, USA; 2007. p. 211–14, doi:http://10.1109/SMI.2007.9.

[8] Sun X, Rosin PL, Martin RR, Langbein FC. Bas-relief generation using adaptive histogram equalization. IEEE Trans Visual Comput Graph 2009;15:642–53. doi:10.1109/TVCG.2009.21.

[9] Belhumeur PN, Kriegman DJ, Yuille AL. The bas-relief ambiguity. Int J Comput Vis 1999;35:33–44. doi:10.1023/A:1008154927611.

[10] Alexa M, Matusik W. Reliefs as images. ACM Trans Graph 2010;29:60:1–7. http://doi.acm.org/10.1145/1778765.1778797.

[11] Cobben J, Elenbaas A. Security feature comprising a perforation pattern. US Patent, Appl. no. 09/297585, filed 20 July 1999.

[12] Cobben J, Augustinus A. Security document with a perforation pattern. US Patent 6,786,513, filed 21 November 2001.

[13] Mitra NJ, Pauly M. Shadow art. ACM Trans Graph 2009;28(5):156:1–7. doi:10.1145/1618452.1618502.

[14] Deussen O, Hiller S, van Overveld C, Strothotte T. Floating points: a method for computing stipple drawings. Computer Graphics Forum 2000; 19(3).

[15] Secord A. Weighted Voronoi stippling. In: Proceedings of the second international symposium on non-photorealistic animation and rendering, NPAR '02, ACM, New York, NY, USA; 2002, p. 37–43, doi:10.1145/508530.508537.

[16] Hiller S, Hellwig H, Deussen O. Beyond stippling—methods for distributing objects on the plane. Comput Graph Forum 2003;22(3):515–22.

[17] Schmaltz C, Gwosdek P, Bruhn A, Weickert J. Electrostatic halftoning. Comput Graph Forum 2010;29:2313–27. doi:10.1111/j.1467-8659.2010.01716.x.

[18] Xu Y, Liu L, Gotsman C, Gortler SJ. Capacity-constrained Delaunay triangulation for point distributions. Comput Graph (Proc Shape Model Int) 2011;35(3):510–6.

[19] Martín D, Arroyo G, Luzón MV, Isenberg T. Example-based stippling using a scale-dependent grayscale process. In: Proceedings of the eighth international symposium on non-photorealistic animation and rendering, NPAR '10, ACM, New York, NY, USA; 2010. p. 51–61, doi:10.1145/1809939.1809946.

[20] Aurenhammer F. Power diagrams: properties, algorithms and applications. SIAM J Comput 1987;16(1):78–96.

[21] Alexa M, Matusik W. Images from self-occlusion. In: Cunningham D, Isenberg T, editors. Workshop on computational aesthetics. Vancouver, Canada: Eurographics Association; 2011. p. 17–24. doi:10.2312/COMPAESTH/COMPAESTH11/017-024.

[22] Mantiuk R, Daly S, Kerofsky L. Display adaptive tone mapping. ACM Trans Graph 2008;27:68:1–10. doi:10.1145/1360612.1360667.

[23] Aurenhammer F, Hoffmann F, Aronov B. Minkowski-type theorems and least-squares clustering. Algorithmica 1998;20(1):61–76.

[24] Hausner A. Simulating decorative mosaics. In: Proceedings of the 28th annual conference on computer graphics and interactive techniques, SIGGRAPH '01, ACM, New York, NY, USA; 2001. p. 573–80, doi:10.1145/383259.383327.

[25] Lloyd SP. Least squares quantization in PCM. IEEE Trans Inf Theory 1982;28:129–37.

[26] Pharr M, Humphreys G. Physically based rendering: from theory to implementation. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 2004.

[27] Kramer TR, Proctor FM, Messina E. The nist rs274ngc interpreter—version 3 (2000).