Computational Aesthetics in Graphics, Visualization, and Imaging (2011)
D. Cunningham and T. Isenberg (Editors)

Images from Self-Occlusion

Marc Alexa[1,3] and Wojciech Matusik[2,3]

[1]TU Berlin [2]MIT [3]Disney Research, Zurich

Figure 1: *Our system uses an arbitrary image as input (left). Then, it produces a surface with embedded holes such that the shading due to occlusion induces the desired image. We fabricate these surfaces using computer-controlled milling machines. We show that aligning holes with image features improves reproduction quality (right) compared to regular hole pattern (center).*

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 we can obtain reproductions that look better than those resulting from regular hole patterns – despite using slightly 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.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Digitizing and scanning

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).

CAe 2011, Vancouver, British Columbia, Canada, August 5 – 7, 2011.
© 2011 ACM 978-1-4503-0908-0/11/0008 $10.00

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 adding small holes into 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 depth. 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. However, we show that it is also beneficial 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 an efficient solution for finding pit centers of the irregular grids. Since regular grids provide the maximum number of holes per area this process necessarily decreases the number of holes. Interestingly, 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 with shading due to self-occlusion depicting a desired image 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 have been explored. For example, Oren and Nayar [ON94] develop a model for Lambertian microfacets. Koenderink et al. [KDDN99] predict an aggregate reflectance of surfaces with spherical concavities. Similarly Mérillou et al. [MDG00] 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 [CMS97, WDB*07, SBS07, SRML09] – 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 [BKY99]). In a recent approach, we have created reliefs for specific light directions, yet not based on shading [AM10].

There have been also a number of methods that attempt to reproduce images on uniform materials. For example, one method [CE99, CA01] 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 [MP09]. They optimize for a volumetric structure that casts up to three different user-specified binary shadows onto planar surfaces. In our case, we optimize for 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 [DHvOS00, Sec02, HHD03, SGBW10]. 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 while the typical stipples have no intensity variation. Second, we have to absolutely enforce the distance between the stipples in order to be able to manufacture the physical surface. Finally, in our case we prefer dense distributions even for mid-tones.

3. Method

As a first step for creating and analyzing surfaces we need to define what we understand as the image generated by the pitted surface. For this, we associate the Voronoi cell of each pit (i.e. its circular region) to the pit and model the gray value in each cell as constant. We assume all pits to have the same radius r and further that no two pits intersect. With these assumptions we can equivalently compute the Voronoi cell of each pit center, which is a significantly simpler computation. Let $P = \{(p_c, p_v)\}$ represent the pits, with p_c the position of

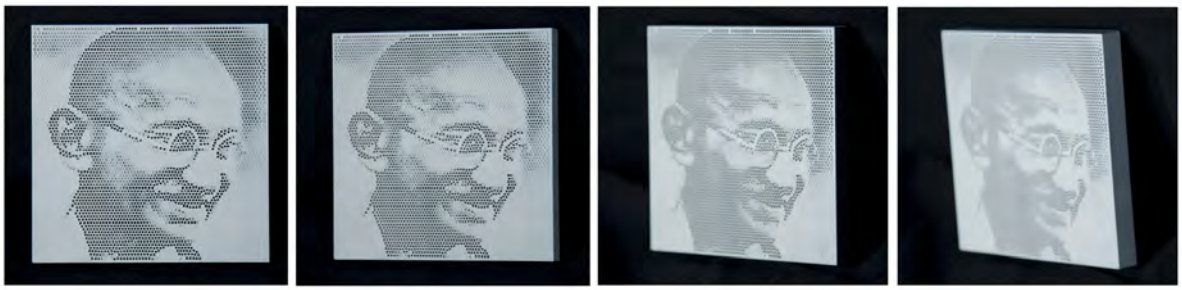

Figure 2: *The photographs of the surface depicting Ghandi fabricated using a 3D printer. The image is clearly visible from different viewpoints.*

the center of the pit p and p_v the associated value, which will depend on the depth of the pit and the associated area.

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. Second, we consider the optimal placement of the pits: while the densest packing for circular elements in the plane is the hexagonal grid, we consider a natural error functional and describe and iterative algorithm to optimize the placement. Last, we describe our simulation procedure to preview surfaces without milling them.

3.1. 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 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 Figure 2). 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 Figure 3. 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 Figure 4). 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.

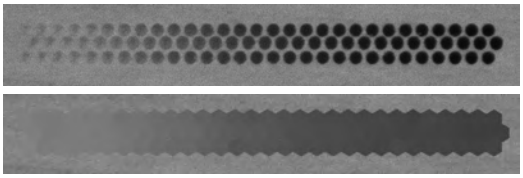

Figure 3: *A drilled calibration pattern in a wood sample (above) and the fitted hexagonal patches (below).*

3.2. 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

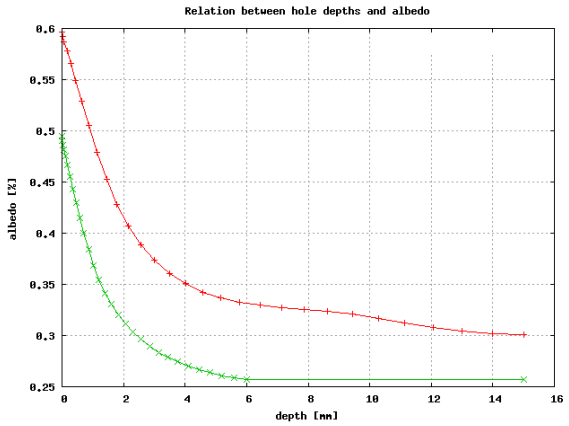

Figure 4: *A measurement-based estimate of average apparent albedo for wood (green) and a 3D printing polymer (red) as a function of pit depth.*

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. [MDK08].

3.3. Pit Placement

Given a surface area, the desired pit radius r, and a minimal distance between holes ($> 2r$), the largest number of pits can be placed using a hexagonal grid for the pit centers. However, it is not clear that the largest number of pits necessarily yields the best reproduction of a given image function $I(x,y)$. And, indeed, we find that the uniform pit distribution of the output image often exposes clearly visible and distracting aliasing artifacts along image edges.

We wish to quantify the difference between the image conveyed by the pitted surface and the image I. For this, we represent the input image $I(x,y)$ with a set of discretely placed samples S. The set S contains elements $s = \{s_c, s_v\}$, where s_c is an image sample coordinate and s_v is the corresponding image value. In practice, we use a grid that is a few times finer than the input raster image, i.e. we represent each pixel in the image by a set of 2 by 2 to 5 by 5 samples. We note that oversampling the input image is useful to suppress aliasing artifacts in the optimization procedure. The idea of using a fine discrete grid for computational purposes is similar to Balzer et al. [BSD09].

Now we can associate each pit p to the discrete representation of the Voronoi cell S in the image, i.e. S_p contains the samples s for which s_c is closest to p_c. Based on this association we define the average error as

$$E = \sum_{p\in P} \frac{1}{|S_p|} \sum_{s\in S_p} \|p_v - s_v\| = \frac{1}{|S|} \sum_{p\in P} \sum_{s\in S_p} \|p_v - s_v\| \quad (1)$$

In order to minimize this error we optimize the pit centers with a weighted variant of Lloyd's method [Llo82]. This algorithm iterates between the definition of the cell associated to each pit and then updating the pit position to the centroid of the cell. The basic observation is that we need to bring each pit into a position so that its value p_v is close to all associated values s_v, i.e. ideally each pit represents a homogeneous area of the image. This idea is reflected in the definition of the two steps below, taking into consideration not only the positions but also the values of the pits and samples.

In the first step, we use the following cost function $c(s,p)$ to associate the samples to a pit position:

$$c(s,p) = \|s_c - p_c\| + \sigma_v \|s_v - p_v\|. \quad (2)$$

The weight σ_v corresponds to the importance of image values compared to the Euclidean distance. Associating all samples to all pits requires $O(|S||P|)$ for a simple scan over all points. We use Dijkstra's algorithm to speed up this process.

In the second step of the algorithm, each pit center is moved towards the weighted centroid of its associated sample points computed in the first step. Assuming the subset S_p contains all image samples in S that are the closest to pit p (based on eq. 2), we calculate the corresponding centroid C_p as follows:

$$C_p = \frac{\sum_{s\in S_p} s_c \exp(-\|p_c - s_c\|^2 \sigma_w)}{\sum_{s\in S_p} \exp(-\|p_c - s_c\|^2 \sigma_w)}, \quad (3)$$

where σ_w is chosen so that it normalizes the image values to the range $[0;1]$. Given a centroid C_p and the corresponding pit p we update each pit position and its value as follows:

$$p_c \leftarrow p_c + \delta(C_p - p_c), \quad (4)$$

where parameter δ controls the step size towards the weighted centroid. We set δ to less than 0.1 in order to improve convergence. This is especially desired when the image sampling grid has low-resolution and the underlying energy function is discontinuous. Furthermore, the pit center update step might place the centers too close to each other for a given pit radius and minimal wall size. We simply disallow these types of updates.

To summarize, the following pseudocode describes the algorithm for irregular pit distribution.

Algorithm 1 $P \leftarrow$ IrregularDistribution (I, r)

```
S ← ComputeImageSamples(I, r)
P ← InitializePits(I, r)
repeat
    For each p ∈ P, S_p ← AssignImageSamples(S_c, p)
    For each p ∈ P, C_p ← ComputeCentroid(S_p, p)
    For each p ∈ P, p_c ← UpdatePit(C_p, p_c)
    For each p ∈ P, p_v ← UpdateValue(S_p)
until P converges
For each p ∈ P, ComputeHoleDepth(S_p, p_v)
```

3.4. 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 [PH04]. 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 the surface patch including a pit and its surrounding area. Thus, in order to assemble the whole simulated image we simple composite sprites corresponding to the appropriate albedo values. We show the results of this image-based synthesis in Section 4.

4. Results

In this section we compare and analyze the regular and irregular distribution of pits. Next we discuss the surface fabrication process. Then we present the examples of fabricated surfaces. Finally we describe the limitations of the method.

4.1. Comparison of Pit Distributions

The final self-occlusion surface is described as a set of pit centers P computed as explained in section 3. We compare the regular (hexagonal) pit distribution to the irregular distribution for one of the examples (Figure 5) using the error functional in eq. 1. The regular distribution uses 6510 pits and results in the average error of 12.05 whereas the irregular pit arrangement utilizes 5213 pits yielding the error of 8.18. As intended the irregular placement minimizes the error along the contrast edges in the input image. We show an additional comparison of regular vs. irregular distribution for font reproduction in Figure 6.

Figure 7 depicts the results of the algorithm for computing the irregular pit distribution. The left image set shows the initial state where the pit centers are still placed as the regular grid. Due to the regular pit placement the positions of the pits are not aligned with the image features leading to higher errors. On the right side we show the state after 400 iterations. In this case the sites are now placed aligned with the image features leading to smaller errors on many edges and visually smooth shapes. This clearly shows that for our *weighted* version of the algorithm the initial hexagonal pattern is not optima and, hence, generally not a steady state.

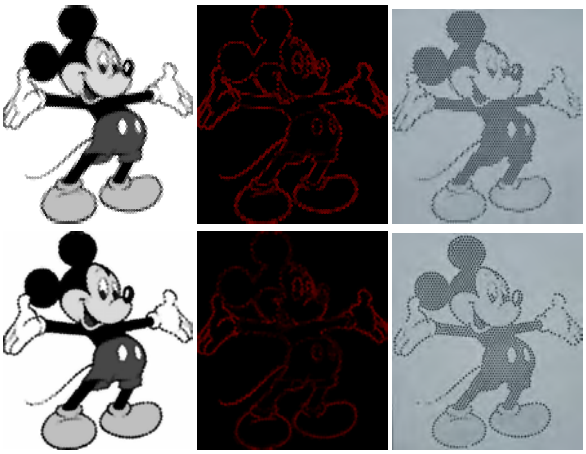

Figure 5: *Uniform hexagonal distribution (upper row) vs. irregular placement (lower row). The second column shows the average error computed for each pit (black – low error value, red – high error value). The last column shows the resulting 3D prints.*

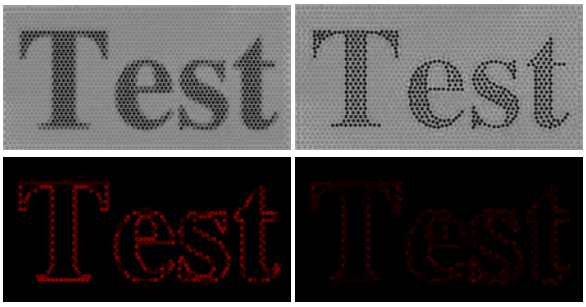

Figure 6: *Comparison of the uniform hexagonal tessellation (left) and the optimized irregular placement (right) for font depiction. The first row shows the results of the simulation. The second row shows the errors for each cell.*

It should be noted though that the method naturally cannot overcome limitations imposed by the sampling frequency as e.g., the two thin black rings in the lower left part of the input image. Such features will still suffer from aliasing regardless of the used method.

4.2. 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 x 146 mm. Moreover, the machine is

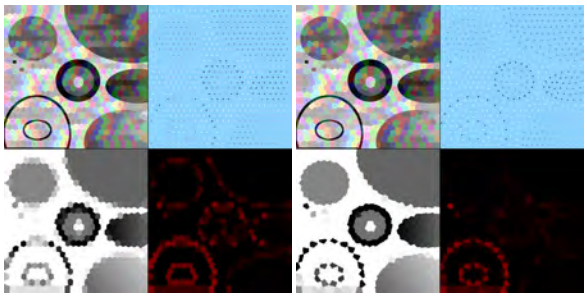

Figure 7: *The initial state (left) and an advanced state (right) of the optimization. Clockwise starting from images in upper left in each set: regions corresponding to each pit, centroid distribution, average error for each region, average region values.*

Figure 8: *Input images are on the left. Both inputs are were used to create optimized patterns. Upper row (Stapelia flower) shows a 3D print, lower row (Print "Day and Night" by M.C. Escher) irregular drilling pattern in wood.*

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) [KPM00] – a standard command language for CNC machine tools. On our machine a typical image of about 10000 pits resolution takes roughly five to seven hours depending on the hole depths. Our milled examples have sizes of about 130 mm x 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 the used 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 x 350 mm. Furthermore, the printer offers excellent precision (e.g., the resolution in x, y, and z direction is 30 microns). 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.

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

4.3. Limitations

Our method has a few limitations. One of the major limitations is the limited contrast (see e.g. Figure 8). 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 percent. We can easily obtain surfaces with higher albedo (90 percent or higher) such as 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. 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 is decreasing when looking at the surface from oblique angles.

5. 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 Figure 9 and earlier examples). We believe

Figure 9: *An image of stone pebbles (left) and the results drilled in wood (center) and 3D printed using an optimized irregular drill pattern (right).*

that this method provides designers, artists, hobbyist with a powerful tool for 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. We also would like to explore more complex optimization strategies for hole placement (e.g., different hole insertion and removal strategies). 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

[AM10] ALEXA M., MATUSIK W.: Reliefs as images. *ACM Trans. Graph. 29* (July 2010), 60:1–60:7.

[BKY99] BELHUMEUR P. N., KRIEGMAN D. J., YUILLE A. L.: The bas-relief ambiguity. *Int. J. Comput. Vision 35* (November 1999), 33–44.

[BSD09] BALZER M., SCHLÖMER T., DEUSSEN O.: Capacity-constrained point distributions: a variant of lloyd's method. *ACM Trans. Graph. 28* (July 2009), 86:1–86:8.

[CA01] COBBEN J., AUGUSTINUS A.: Security document with a perforation pattern. US Pat 6,786,513, filed November 21, 2001, 2001.

[CE99] COBBEN J., ELENBAAS A.: Security feature comprising a perforation pattern. US Pat. Appl. 09/297585, filed July 20, 1999, 1999.

[CMS97] CIGNONI P., MONTANI C., SCOPIGNO R.: Computer-assisted generation of bas- and high-reliefs. *J. Graph. Tools 2* (December 1997), 15–28.

[DHvOS00] DEUSSEN O., HILLER S., VAN OVERVELD C., STROTHOTTE T.: Floating points: A method for computing stipple drawings. *Computer Graphics Forum 19*, 3 (Aug. 2000).

[HHD03] HILLER S., HELLWIG H., DEUSSEN O.: Beyond stippling - methods for distributing objects on the plane. *Computer Graphics Forum 22*, 3 (Sept. 2003), 515–522.

[KDDN99] KOENDERINK J. J., DOORN A. J. V., DANA K. J., NAYAR S.: Bidirectional reflection distribution function of thoroughly pitted surfaces. *International Journal of Computer Vision 31*, 2–3 (April 1999), 129–144.

[KPM00] KRAMER T. R., PROCTOR F. M., MESSINA E.: The nist rs274ngc interpreter - version 3.

[Llo82] LLOYD S. P.: Least squares quantization in pcm. *IEEE Transactions on Information Theory 28* (1982), 129–137.

[MDG00] MÉRILLOU S., DISCHLER J.-M., GHAZANFARPOUR D.: A brdf postprocess to integrate porosity on rendered surfaces. *IEEE Transactions on Visualization and Computer Graphics 6*, 4 (2000), 306–318.

[MDK08] MANTIUK R., DALY S., KEROFSKY L.: Display adaptive tone mapping. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers* (New York, NY, USA, 2008), ACM, pp. 1–10.

[MP09] MITRA N. J., PAULY M.: Shadow art. *ACM Transactions on Graphics 28*, 5 (2009), 1–7.

[ON94] OREN M., NAYAR S. K.: Generalization of lambert's reflectance model. In *Proceedings of SIGGRAPH 94* (July 1994), Computer Graphics Proceedings, Annual Conference Series, pp. 239–246.

[PH04] PHARR M., HUMPHREYS G.: *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[SBS07] 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* (Washington, DC, USA, 2007), IEEE Computer Society, pp. 211–214.

[Sec02] SECORD A.: Weighted voronoi stippling. In *NPAR 2002: Second International Symposium on Non Photorealistic Rendering* (June 2002), pp. 27–43.

[SGBW10] SCHMALTZ C., GWOSDEK P., BRUHN A., WEICKERT J.: Electrostatic halftoning. *Computer Graphics Forum 29* (2010), 2313–2327.

[SRML09] SUN X., ROSIN P. L., MARTIN R. R., LANGBEIN F. C.: Bas-relief generation using adaptive histogram equalization. *IEEE Transactions on Visualization and Computer Graphics 15* (July 2009), 642–653.

[WDB*07] WEYRICH T., DENG J., BARNES C., RUSINKIEWICZ S., FINKELSTEIN A.: Digital bas-relief from 3d scenes. *ACM Trans. Graph. 26*, 3 (2007), 32.