

Fabricating Reflectors for Displaying Multiple Images

KAISEI SAKURAI, DWANGO Co., Ltd., Dwango CG Research

YOSHINORI DOBASHI, Hokkaido University and Dwango CG Research

KEI IWASAKI, Wakayama University and Dwango CG Research

TOMOYUKI NISHITA, Dwango CG Research and Hiroshima Shudo University

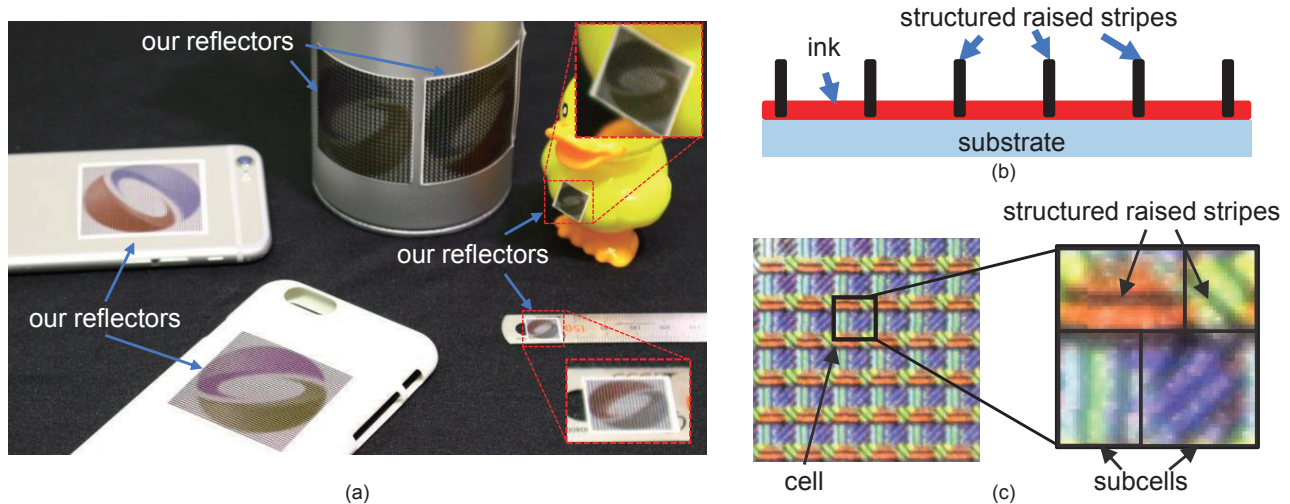


Fig. 1. (a) Examples of our reflectors printed on everyday personal objects: a pen holder, smartphone cases, toy ducks, and a ruler. The reflectors on the smartphone cases are directly printed by a standard UV printer. Other reflectors are printed on stickers and pasted onto the objects. All the reflectors are designed to display SIGGRAPH logo with different colors depending on the viewing direction (see Fig. 6 for input images). (b) Microstructure of the printed pattern consisting of raised stripes on an ink layer. (c) Close-up photograph of printed patterns.

A great deal of attention has been devoted to the fabrication of reflectors that can display different color images when viewed from different directions not only in industry but also for the arts. Although such reflectors have previously been successfully fabricated, the number of images displayed has been limited to two or they suffer from ghosting artifacts where mixed images appear. Furthermore, the previous methods need special hardware and/or materials to fabricate the reflectors. Thus, those techniques are not suitable for printing reflectors on everyday personal objects made of different materials, such as name cards, letter sheets, envelopes, and plastic cases. To overcome these limitations, we propose a method for fabricating reflectors using a standard ultraviolet printer (UV printer). UV printer can render a specified 2D color pattern on an arbitrary material and by overprinting the printed pattern can be raised, that is, the printed pattern

Authors' addresses: Kaisei Sakurai, DWANGO Co., Ltd., Dwango CG Research, kaisei_sakurai@dwango.co.jp; Yoshinori Dobashi, Hokkaido University, Dwango CG Research, doba@ime.ist.hokudai.ac.jp; Kei Iwasaki, Wakayama University, Dwango CG Research, iwasaki@sys.wakayama-u.ac.jp; Tomoyuki Nishita, Dwango CG Research, Hiroshima Shudo University, nishita@shudo-u.ac.jp.

Permission to make digital or hard copies of all or part 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 bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Association for Computing Machinery.

0730-0301/2018/8-ART158 \$15.00

<https://doi.org/10.1145/3197517.3201400>

becomes a microstructure having color and height. We propose using these microstructures to formulate a method for designing spatially varying reflections that can display different target images when viewed from different directions. The microstructure is calculated by minimizing an objective function that measures the differences between the intensities of the light reflected from the reflector and that of the target image. We show several fabricated reflectors to demonstrate the usefulness of the proposed method.

CCS Concepts: • **Computing methodologies** → *Image manipulation*; *Reflectance modeling*;

Additional Key Words and Phrases: digital fabrication, reflector, ultraviolet printer, spatially varying anisotropic BRDF

ACM Reference Format:

Kaisei Sakurai, Yoshinori Dobashi, Kei Iwasaki, and Tomoyuki Nishita. 2018. Fabricating Reflectors for Displaying Multiple Images. *ACM Trans. Graph.* 37, 4, Article 158 (August 2018), 10 pages. <https://doi.org/10.1145/3197517.3201400>

1 INTRODUCTION

To address the demand for increasing variation in industrial design, a number of techniques for displaying images have been developed, such as reflectors [Regg et al. 2010; Snelgrove et al. 2013; Weyrich et al. 2007], relief sculptures [Alexa and Matusik 2010; Belhumeur et al. 1999; Mitra and Pauly 2009; Schüller et al. 2014; Song et al. 2007; Sýkora et al. 2014], and lenses [Baran et al. 2012; Papas et al. 2012;

Yue et al. 2012]. In recent years, several methods have emerged that improve the ability of metallic reflectors to display various images [Pjanic and Hersch 2015a,b]. These methods enable the fabrication of metallic reflectors that can display specified images by specular reflection. However, in these methods, the number of images to be displayed is limited to two since a phenomenon known as ghosting, where images other than the specified image are displayed, arises.

One additional but important issue with the previous methods is how the reflector is fabricated; these often need special devices and/or materials for fabrication. Thus, the benefits of the method cannot easily be obtained unless one has access to such devices/material. This prevents many interesting reflectors becoming widespread. If the reflectors can be printed easily by anyone, they can be used for different personal purposes. For example, we can print reflectors on letters, name cards, envelopes, posters, or (curved) plastic cases, etc., for decoration, artwork, or advertisement. The previous methods do not allow us to use the techniques for these personal purposes with a reasonable printing cost.

Our goal is to address the above problems. We use a standard ultraviolet (UV) printer, which enables anyone to print our reflectors on their own everyday objects. A UV printer is capable of printing UV curable ink on different types of material, including paper, plastic, metal, transparent sheet, etc. It can also print on curved surfaces. UV printers are usually available via a printing agency or one can simply buy one. Our method provides the user with a tool to compute the printing pattern to be fed to the UV printer. Then, the printed reflector displays multiple images specified by the user toward different viewing directions.

To achieve this goal, we make use of the ability of UV printers to fabricate tiny height fields by overprinting. The optimal microstructure is computed and represented as a height field printable by the UV printer. In fact, with our method, the number of images displayed with less ghosting artifacts can be increased. Our method renders spatially-varying diffuse reflections on the reflector by fabricating high-resolution microstructures in order to produce the desired anisotropic reflections. The microstructure is computed by minimizing an energy function representing the sum of the differences between the desired intensities and the simulated results. This problem is hard to solve in general because this energy function is a high-dimensional, nonlinear function of the lighting conditions, viewing directions, and the spatially varying BRDF (SVBRDF) of the reflector. We focus here on a SVBRDF that minimizes the energy function, assuming fixed lighting conditions and viewing directions.

The microstructure on our reflector is designed in the form of a stripe pattern, as shown in Fig. 1(c). The stripe pattern consists of two parts: an ink layer and small-scale vertical walls called *structured raised stripes* (SRSs), as shown in Fig. 1(b). We decompose the SVBRDF into spectral and directional components, which correspond to the ink layer and the SRSs, respectively. The directional component is derived from the SRSs that occlude the reflected light. The spectral component, resulting from the combination of the printed inks, controls the color of the SVBRDF.

Figure 1(a) shows an example of our reflectors printed on different objects. The reflectors in this figure are designed to display the SIGGRAPH logo with four different colors depending on the viewing

direction. Fig. 1(c) shows a close-up view of the reflector where we can observe the SRSs (black line segments) and the inks making up the reflector. In order to fabricate the reflector, we use a UV printer. UV curable ink can be stacked by overprinting. The SRSs are fabricated by overprinting black lines over the ink layer.

The contributions of this paper are summarized as follows.

- Our reflectors can be easily fabricated by printing patterns using a standard UV printer, which is readily accessible through commercial printer services. Anyone can easily reproduce our reflectors.
- Our reflector does not require additional equipments (e.g. a lens for lenticular and parallax barriers), nor electrical power (e.g. lightfield displays), to display multiple color images.
- We propose a novel method for designing microstructural stripe patterns to display multiple color images with minimal ghosting effects and better clarity.

2 RELATED WORK

Pjanic and Hersch [2015a] displayed a single image using a metallic reflector that had different colors when viewed from two different directions. This was achieved by printing a grid pattern, or a collection of horizontal and vertical line segments on the reflector. However, multiple images with sufficient quality could not be displayed using this method, because one image appeared slightly in the other. The authors [Pjanic and Hersch 2015b] also proposed a method using a metallic reflector, where two colored images are printed on it. Although the purpose of this method is to hide one image in the other, the number of displayable images is limited to two.

Snelgrove et al. [2013] proposed a method for displaying an image with different colors, depending on the elevation angle of the viewing direction. This method is based on the idea of a parallax barrier by showing a portion of the diffuse colored surface. The difference between the gradient of the target image and that of the presented image is minimized in order to preserve the edges of the target image. For this reason, the images are displayed with undesirable colors. Furthermore, the viewing direction in this method is limited. In our method, which can be considered to be a generalization of his method, the viewing direction can be chosen arbitrarily.

The work most closely related to ours is that by Sakurai et al. [2017], who use a similar microstructure to ours. However, they do not provide a method to determine the 'optimal' microstructure; they manually determine a good-looking microstructure via a trial and error process. This may be fine for artwork but the technical contribution in their paper is minimal. Furthermore, their fabricated reflector often produced ghosting artifacts. Our approach is fundamentally different; we propose a method to *automatically compute* the optimal microstructure comprising structured raised stripes.

We can use different optical phenomena to display the images, such as self-shadows [Bermano et al. 2012], diffuse reflections [Alexa and Matusik 2010], refractions [Papad et al. 2012], shadows [Mittra and Pauly 2009], and moiré imagings [Hersch and Chosson 2004]. Specular holograms can display a series of images by specular reflection with a collection of grooves [Brand 2011; Regg et al. 2010].

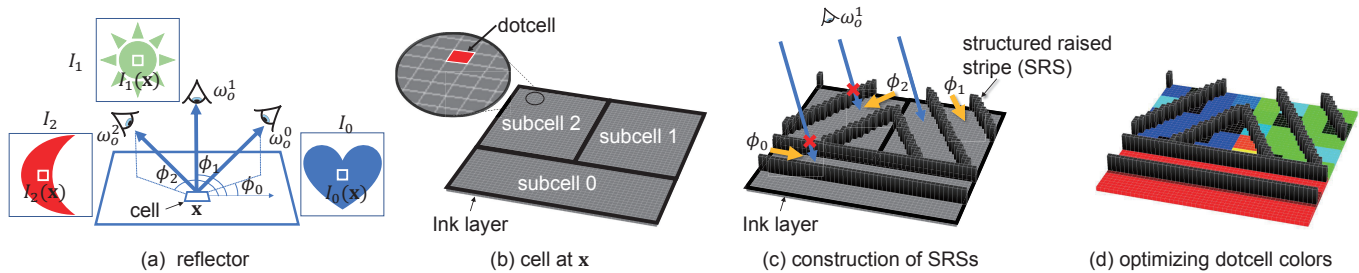


Fig. 2. The basic idea of our method for the case in which $N = 3$. (a) We designed a SVBRDF $f(\mathbf{x}, \omega_o)$ to display the target images I_0 , I_1 , and I_2 for the viewing directions ω_o^0 , ω_o^1 , and ω_o^2 , respectively. ϕ_0 , ϕ_1 , and ϕ_2 are the azimuthal angles for ω_o^0 , ω_o^1 , and ω_o^2 , respectively. (b) The cell at \mathbf{x} is subdivided into three subcells, and each subcell corresponds to $f(\mathbf{x}, \omega_o^0)$, $f(\mathbf{x}, \omega_o^1)$, and $f(\mathbf{x}, \omega_o^2)$. (c) The directions of the SRSs (shown as black walls) are aligned to ϕ_0 , ϕ_1 , and ϕ_2 . The SRSs block the light from ω_o^0 incident on subcells 0 and 2, making only subcell 1 is visible to viewing direction ω_o^1 . (d) Finally, the colors of the dotcells are determined by optimization to render the three colors of the pixel for the three target images at \mathbf{x} .

These methods, however, cannot take into account the reflectance of the colors.

Multiple images can also be displayed by controlling the BRDF. Lan et al. [2013] developed a method for producing a colored reflector that presents an arbitrary BRDF. The limitation of that method is that the presented BRDF depends on the triangular mesh of the reflector. Although this could be used to present colored images, the data size of the triangular mesh is too large to be printed. Matusik et al. [2009] proposed a method for printing a BRDF with a combination of inks, with which a BRDF in a printable range is derived from a given BRDF; however, it is difficult to display multiple colored images by reflection because the method is limited to isotropic BRDFs and it is impossible to change the colors at a given position. There are also methods that use microstructures to control the intensity of the reflectance [Glasner et al. 2014; Levin et al. 2013]. These approaches present anisotropic reflection, but are not suitable for colored reflections; therefore, the method is limited to grayscale images only.

Light field displays have also an ability to display different images depending on view directions. Some methods use stacks of liquid-crystal displays (LCDs), each of which displays its own basis image [Lanman et al. 2011; Wetzstein et al. 2011, 2012]. These methods can render images with high brightness and contrast but ghosting artifacts could be produced due to the fact that all the layers are visible. A parallax barrier display can be constructed by using two LCDs [Lanman et al. 2010; Perlin et al. 2000]. The front panel contains a uniform grid of slits or pinholes, through which the viewer sees a subset of pixels in the rear panel depending on the viewer location. However, these devices need some gaps between the LCD layers and they require electricity and special hardware setups. Therefore, these are not suitable for our purpose, printing reflectors on posters or stickers, etc.

3 BASIC IDEA

Our goal is to fabricate a reflector using a SVBRDF, f , by printing stripe patterns on an ink layer so that the reflector can display N target images viewed from N directions ω_o^k ($k = 0, \dots, N-1$) specified by the user. We assume that all the target images have the same resolution, and the surface of the reflector is subdivided

into grid cells with the same resolution. Therefore, each square cell corresponds to a pixel in each of the target images. We also assume that the viewpoints and the light source are located sufficiently far from the reflector so that the directions from each cell to the viewpoint and to the light source can be considered to be parallel, and the SVBRDF, f , is assumed to be the average reflectance over the cell.

We first explain the basic idea of the method using Fig. 2 where the number of target images N is three. Figs. 2(b) through (d) illustrate the construction of a microstructure used to display three colors at \mathbf{x} in Fig. 2(a). Fig. 2(b) shows the cell at \mathbf{x} , which is a collection of a number of grid cells, each of which corresponds to a dot for the printer. We refer to these cells as dotcells. Our method subdivides the cell into subcells. These subcells correspond to each target image. SRSs are then generated by overprinting with black ink as shown in Fig. 2(c). The direction of SRSs in each subcell k is aligned to the viewing direction ω_o^k . The height of the SRSs and the gaps between them are determined so that each subcell k is visible from the corresponding viewing direction ω_o^k only. In Fig. 2(c), for example, rays from viewing direction ω_o^1 can reach subcell 1 but other rays reaching subcells 1 and 2 are blocked by the SRSs. Finally, the colors of the dotcells are determined by optimization to replicate each target image.

Problem Definition. SVBRDF f is a function of the cell \mathbf{x} , the incident direction ω_i , and the outgoing direction ω_o , that is, $f(\mathbf{x}, \omega_i, \omega_o)$. To make the design feasible, we design f for incident light with ω_i parallel to the normal \mathbf{n} of the cell with unit intensity, though our method can display multiple target images in general illumination cases as shown in Fig. 1. Since incident light parallel to \mathbf{n} is not occluded by the SRSs and the cosine factor calculated by the dot product of ω_i and \mathbf{n} is simplified to 1, the reflected radiance from cell \mathbf{x} illuminated by unit intensity then depends on two parameters, i.e., $f(\mathbf{x}, \omega_o)$. The reflected radiance $L(\mathbf{y}_j, \omega_o)$ from the j -th dotcell \mathbf{y}_j in \mathbf{x} is calculated from the geometric structure G of SRSs in the cell as:

$$L(\mathbf{y}_j, \omega_o) = v(\mathbf{y}_j, \omega_o; G) \rho_d(\mathbf{y}_j), \quad (1)$$

where $v(\mathbf{y}_j, \omega_o; G)$ is a binary visibility function that returns 1 if a ray from \mathbf{y}_j in direction ω_o does not intersect the SRSs G , and

returns 0 otherwise. $\rho_d(\mathbf{y}_j) \in \mathbb{R}^3$ is the diffuse reflectance at \mathbf{y}_j , which is calculated from the color of the ink printed on \mathbf{y}_j . Since the reflected radiance from cell \mathbf{x} is the average of those from the dotcells, f is calculated from G and ρ_d as follows:

$$f(\mathbf{x}, \omega_0) = \frac{\sum_{j=1}^M v(\mathbf{y}_j, \omega_0; G) \rho_d(\mathbf{y}_j)}{M}, \quad (2)$$

where M is the number of dotcells in the cell.

For each cell \mathbf{x} , we define the following objective function

$$\epsilon(\mathbf{x}) = \sum_{k=0}^{N-1} \|f(\mathbf{x}, \omega_0^k) - p_k(\mathbf{x})/N\|, \quad (3)$$

where $p_k \in \mathbb{R}^3$ is the pixel intensity associated with cell \mathbf{x} in the k -th target image. The division by N is necessary since $f(\mathbf{x}, \omega_0^k)$ is proportional to the area of the corresponding subcell, which decreases as N increases.

By substituting Eq. (2) into Eq. (3), the objective function for cell \mathbf{x} can be represented by the following matrix:

$$\arg \min_{G, C(\mathbf{x})} \|V(G) \cdot C(\mathbf{x}) - P(\mathbf{x})\|, \quad (4)$$

where $V(G)$ is an $N \times M$ visibility matrix given by:

$$V(G) = \frac{1}{M} \begin{pmatrix} v(\mathbf{y}_1, \omega_0^0; G) & \cdots & v(\mathbf{y}_M, \omega_0^0; G) \\ \vdots & \ddots & \vdots \\ v(\mathbf{y}_1, \omega_0^{N-1}; G) & \cdots & v(\mathbf{y}_M, \omega_0^{N-1}; G) \end{pmatrix}, \quad (5)$$

$C(\mathbf{x})$ and $P(\mathbf{x})$ are $M \times 3$ and $N \times 3$ matrices, respectively.

$$C(\mathbf{x}) = \begin{pmatrix} \rho_d(\mathbf{y}_1)^T \\ \vdots \\ \rho_d(\mathbf{y}_M)^T \end{pmatrix}, P(\mathbf{x}) = \begin{pmatrix} p_1(\mathbf{x})^T/N \\ \vdots \\ p_N(\mathbf{x})^T/N \end{pmatrix}. \quad (6)$$

4 DESIGNING THE SVBRDF

To design a SVBRDF to display N target images, we calculate the geometric structure G of the SRSs and the matrix C that minimizes the objective function in Eq. (4). Although the objective function is represented by a simple matrix, it is difficult to optimize this since even if an optimal visibility matrix V^{opt} is obtained, finding the geometric structure $G^{opt} = V^{-1}(V^{opt})$ is infeasible due to the extensive solution space. There are more than 2^M possible patterns of the geometric structures in the cell, resulting in a prohibitive computation time to find the optimum one. On the other hand, if G is determined, it is easy to compute the visibility matrix V by ray casting.

To tackle this, we design the geometric structures G that determine the visibility matrix V in the first step, then in the second step, C is calculated to minimize Eq. (4) for each cell \mathbf{x} . The first step jointly optimizes the geometric structures of all the SRSs in the cell. The direction of the SRSs printed on the k -th subcell is the same and set to be parallel to the azimuthal angle of ω_0^k . Other geometric parameters of the SRSs such as their height and the gaps between them are calculated to minimize the sum of the objective functions representing the amount of light occluded by the SRSs, the ghosting effects, and the deviation of the reflected radiances described in Sec. 4.1.

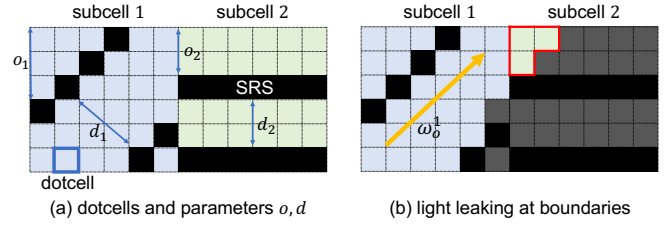


Fig. 3. (a) Dotcells and the parameters for the SRSs. The offsets, o_1 and o_2 , and the gaps between the SRSs, d_1 and d_2 , are shown in subcell 1 (dotcells shaded in blue) and subcell 2 (dotcells shaded in green). The dotcells shaded in black represent the SRSs. (b) When the cell is viewed from direction ω_0^1 (i.e. the azimuthal angle of ω_0^1 is aligned to the direction of the SRSs in subcell 1), the dotcells in subcell 1 are expected to be visible, and those in subcell 2 are not visible (dotcells shaded in gray are shadowed/masked by SRSs). However, reflected light from some dotcells in subcell 2 (surrounded by red lines) can reach the viewpoint. This light leakage results in ghosting effects.

4.1 Geometric Structure of the SRSs

The geometric structure of the SRSs printed on the k -th subcell depends on the height of the SRSs, h_k , the gap between them, d_k , and the offset, o_k , which is the vertical distance of the nearest SRS from the upper left dotcell as shown in Fig. 3(a). The SRSs depend on the area of the k -th subcell as well, which is controlled by the ratio of the area of the k -th subcell to that of the entire cell, a_k . Of these parameters, our method uses a common height parameter h for all the subcells to reduce the number of parameters. We define the parameter set of the SRSs in the k -th subcell as $G_k = \{a_k, d_k, o_k\}$, and define $G = \{h, G_0, \dots, G_{N-1}\}$.

We determine the parameter set G by minimizing an objective function that measures the ghosting effects. As shown in Fig. 3, since the masking effects at the dotcells in the k -th subcell can be affected by the SRSs in neighboring subcells (especially dotcells at the boundary of two subcells), our method solves the minimization problem simultaneously for the whole parameter set G . The same eight cells are placed around the cell to take into account occlusions from SRSs in neighboring subcells. The same parameter set G is used for all the cells (i.e. the SRSs in each cell are identical), making fabrication simple and optimization computationally efficient.

4.1.1 Objective function for G . To display multiple target images faithfully, we want the reflector (1) to reflect the light as strongly as possible for better visibility, (2) to reduce the ghosting effects, and (3) to display each of the N target images with as similar reflected radiances as possible. The objective function $E(G)$ comprises three terms that correspond to the above three demands and is given by:

$$E(G) = w_o E_o(G) + w_g E_g(G) + w_s E_s(G), \quad (7)$$

where E_o , E_g , and E_s are the occlusion term, the ghosting effect term, and the standard deviation term, respectively. w_o , w_g , and w_s are user-specified weights for each term, respectively. To formulate the three terms, we define a function $D_k(\omega_0; G)$ that represents the ratio of dotcells in the k -th subcell visible from direction ω_0 as:

$$D_k(\omega_0; G) = \frac{\sum_{j \in J_k} v(\mathbf{y}_j, \omega_0; G)}{M}, \quad (8)$$

where J_k is the set of dotcell indices in the k -th subcell.

To make the reflected radiances from the cells for N directions large, the occlusion term E_o is calculated as the sum of the amount of light occluded by the SRSs using the following equation:

$$E_o(G) = \sum_{k=0}^{N-1} (1 - D_k(\omega_o^k; G)). \quad (9)$$

The ghosting effect viewed from the direction ω_o^k is caused by light leaking from subcells other than the k -th subcell and thus the ghosting effect for ω_o^k can be measured by the sum of D_l except for $l = k$ as follows:

$$\sum_{l=0, l \neq k}^{N-1} D_l(\omega_o^k; G). \quad (10)$$

The ghosting effect term E_g is calculated using the following equation:

$$E_g(G) = \sum_{k=0}^{N-1} \sum_{l=0, l \neq k}^{N-1} D_l(\omega_o^k; G). \quad (11)$$

To make the reflected radiances of the N target images as similar as possible, our method uses the standard deviation term E_s :

$$E_s(G) = \sqrt{\frac{1}{N} \sum_{k=0}^{N-1} (D_k(\omega_o^k; G) - \bar{D})^2}, \quad (12)$$

where \bar{D} is the average of D_k , calculated by $\bar{D} = \frac{\sum_{k=0}^{N-1} D_k(\omega_o^k; G)}{N}$. Our method uses the standard deviation instead of the variance in order to equalize the degree for D to that of the other two terms E_o and E_g (the degree of the variance in D is two, while that for E_o and E_g is one).

Algorithm 1 shows the procedure for evaluating the energy function. First the cell is subdivided into subcells according to the parameter a_k in G using the method described in the next section. $D_l(\omega_o^k; G)$ is then evaluated by casting rays from each of the dotcells in subcell l to the viewing direction ω_o^k for the visibility function v in Eq. (8).

4.1.2 Cell Subdivision. In the optimization process, each cell is subdivided according to the ratio a_k . We prefer to make the minimum edge length of the subcell not too small, since the subcell size can be smaller than the gap between the SRSs, which makes it difficult to draw the line segments for the SRSs using a UV printer.

To avoid this, we want to maximize the smallest length of the edge of the subcells. Fig. 4 illustrates our subdivision algorithm for $N = 4$. Our algorithm first calculates $a_0 + a_2$ and splits the cell into two groups using the ratio $\frac{a_0 + a_2}{\sum_{i=0}^3 a_i}$. Next, each of the two groups are further subdivided in the same way but we choose the direction that is perpendicular to the longer edge of the subcell for the subdivision. These processes are repeated recursively.

4.1.3 Optimization of $E(G)$. The parameter set G is calculated by solving the minimization problem for $E(G)$. The objective function $E(G)$ is minimized by using a genetic algorithm. We follow the standard approach [Goldberg 1989]. Each parameter is quantized and converted into a corresponding binary bit string. The range of

Algorithm 1 Evaluation of $E(G)$. $G = \{h, G_0, \dots, G_{N-1}\}$ is the parameter set for all subcells.

```

1: procedure COMPUTE $E(G)$ 
2:   subdivide cell into subcells according to  $a_k$  in  $G$ 
3:   for  $l = 0$  to  $N - 1$  do
4:     for  $k = 0$  to  $N - 1$  do
5:       calculate  $D_l(\omega_o^k; G)$  by ray casting
6:   calculate  $E_o(G)$ ,  $E_g(G)$ ,  $E_s(G)$ , and  $E(G)$ 
7:   return  $E$ 

```

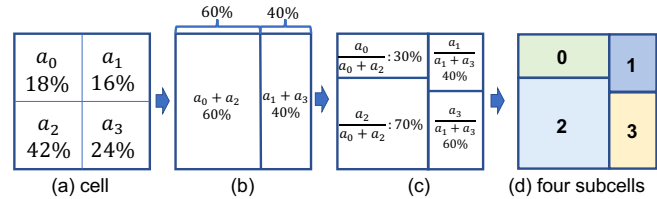


Fig. 4. Examples of subdivision of a cell into four subcells according to (a) the ratio of the area of each subcell to the area of the cell, a_i . (b) two groups are partitioned vertically based on the sum of the ratio between $a_0 + a_2$ and $a_1 + a_3$. (c) each group is further partitioned based on the ratios but the split direction is selected perpendicular to the direction of the longer edge of the subcell. (d) each group is subdivided according to the ratio a_i .

each parameter is specified by the user. The bit strings for all the parameters are connected together to form a single longer bit string that is used as the gene of an individual. Initially, the GAs generate n individuals using random numbers, where n is specified by the user. The GAs iteratively generate a new set of n individuals based on a fitness function that evaluates the quality of each individual. We use the inverse of our objective function, $1/(E + 1)$, as the fitness function so that the maximum is one. New individuals are generated through two genetic operators, one called a crossover operator and the other a mutation operator. We use a two-point crossover. We also employ a so-called elitist selection strategy, by which the best individual at a certain iteration is carried over to the next iteration. We chose these genetic operators experimentally. Although there are different operators such as a one-point crossover, we have not observed any significant differences in the convergence of the objective function. The elitist selection, however, is important to increase the convergence speed. For more details about GAs, there are many good textbooks such as the one by Goldberg [1989].

The search process terminates if the fitness function becomes smaller than a specified threshold ϵ_{GA} or the number of iterations exceeds a specified number n_{max} . $\epsilon_{GA} = 0.999$ and $n_{max} = 500$ worked fine for all cases. We used the typical values mentioned in [Goldberg 1989] for the other parameters of the GAs.

4.2 Diffuse reflectance

Once the geometric structures for G are obtained, the visibility matrix $V(G)$ is calculated by ray-casting. Our method then determines C , which comprises the diffuse reflectance for each dotcell by solving the minimization problem for Eq. (4). A naive solution, such as the least squares method or the pseudoinverse of V , cannot be

applied to obtain C due to the two constraints for printing: non-negativity and the smoothness of the ink. Since each component of C is the diffuse reflectance of each dotcell, C should be non-negative. In addition, the printer cannot print significantly different colors in neighboring dotcells since the colors of the inks are created using the dithering method and the error diffusion method.

Based on these two constraints, our method employs nonsmooth nonnegative matrix factorization (nsNMF) proposed by Pascual-Montano et al. [2006]. nsNMF approximately decomposes P into the product of three non-negative matrices VSC . S is a smoothing $M \times M$ matrix given by $S = (1 - \theta)I + \theta J$, where θ is a parameter controlling the smoothness, I is the $M \times M$ identity matrix, and all the elements of $M \times M$ matrix J are equal to $1/M$. C is a $M \times 3$ matrix. The original algorithm for nsNMF iteratively updates both V and C to decompose the input matrix P . Since the visibility matrix V is fixed in our case, we only update C . For the smoothing matrix, we use $\theta = 0.1$ for all the examples shown in this paper. When we used larger values for θ , C was too blurry and we could not obtain good results.

We have further found an additional issue when we actually fabricate a reflector with the colors computed by the above method. The method should theoretically work fine but we find that the colors on the actual reflector became darker than expected. This is due to the fact that the printed inks in neighboring dotcells are mixed, resulting in decreased reflectances. To avoid this, we apply an additional averaging process to the colors of the dotcells. That is, we subdivide a cell into regular groups of $m \times m$ dotcells and compute the average color for each group. We use $m = 3$ for all the examples shown in this paper. By incorporating this averaging process into the nsNMF algorithm, we can obtain much better results. Our final procedure to compute C is:

- (1) initialize C with random numbers ranging from zero to one.
- (2) update C by $C_{ij} = C_{ij} * ((VS)^T P)_{ij} / ((VS)^T (VS)C)_{ij}$, where the subscript ij indicates the (i, j) element of the corresponding matrix.
- (3) compute the average colors of the groups of $m \times m$ dotcells and substitute the average colors into the corresponding elements of C .
- (4) clamp each element of C between zero and one.

The above procedure is repeated until the energy is less than a specified threshold or the specified number of iterations is reached. To update C in step (2), we simply follow the method described in [Pascual-Montano et al. 2006].

5 RESULTS

In this section, we present the simulated and fabricated results of our method and compare them to previous work. The simulated results of the reflector are visualized by rendering each dotcell with the evaluated value of f in Eq. (2) taking into account the occlusion due to the SRSs.

The UV printer used for fabrication is a Roland VersaUV LEF-12, which is an inkjet printer that propels droplets of UV curable inks from print head nozzles and emits UV light to dry them. The printer resolution is 720 dots per inch (dpi) and one dot is



Fig. 5. Photographs of the fabricated reflector displaying two different images. The input images are shown in the insets.

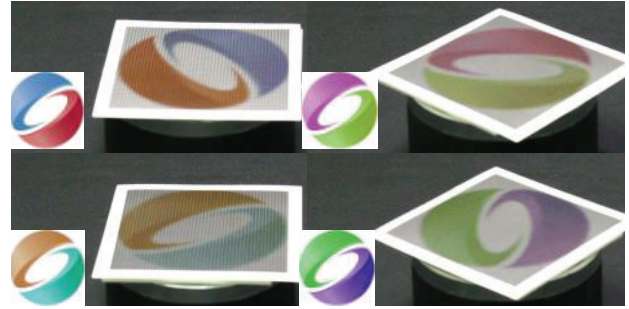


Fig. 6. Color changing effect. Photographs of the fabricated reflector displaying the SIGGRAPH logo with four different color variations. The input images are shown in the inset.

25.4/720=0.03527mm. The SRSs and the ink layer are printed separately. We first print the ink layer on a substrate, then fabricate the SRSs by overprinting black line segments on the ink layer. We found experimentally that the width of the black lines are larger than the size of the dotcell. We measured the actual size of the black lines and take this into account in computing the SRSs. Our method calculates the colors of the reflector in the RGB color space while the UV printer uses CMYK colors. We used GRACoL2006 Coated (ISO 12647-2:2004) for conversion from RGB to CMYK.

We used a standard PC with Intel Core i7-5820K 3.30Ghz (CPU) and 32.0GB RAM for creating the following examples. The computation of the SRSs usually finishes within one minute. However, the computation for the diffuse reflectances of the dotcells takes a few hours. The computation time mainly depend on the size of the cell and the longest computation time among our examples was three hours. All the reflectors in our experiments were fabricated in 3 to 4 hours.

5.1 Experimental Results

This section reports results of some experiments we made to investigate the ability and characteristics of our method.

Displaying two different images. Fig. 5 shows an example of a reflector displaying two different images; the image changes depending on the angle of rotation. We set the two azimuthal angles for Figs. 5(a) and (b) to zero and 90 degrees, respectively. The size of this reflector is $108.37 \times 108.37 \text{ mm}^2$, and the resolution of the target image is 128×128 pixels (each cell consists of 24^2 dotcells). Although the input images are completely different, the reflector can show each image clearly, and the ghosting effect is negligible.

Color changing effect. Fig. 6 shows an example of a color changing effect; the hue of the SIGGRAPH logo changes depending on angle of rotation. Four input images whose hues are 0, 90, 180, and 270 degrees are used to fabricate the reflector. Although the previous method [Pjanic and Hersch 2015a] achieved the color changing effect, the number of colors is limited to two. Our method can produce a color changing effect with more than three colors. The size of this reflector is $73.92 \times 73.92 \text{ mm}^2$, and the resolution of the target image is 25×25 pixels (each cell consists of 48^2 dotcells). Depending on the viewing direction, mixed colors appear on the logo.

Validation of Our Prediction Model. We measured the reflectances of the fabricated reflector and compared them with the simulated results. The comparison is done as follows using a KONICA MINOLTA CS-150 that can measure the intensity of an object. This device measures the intensity in XYZ color space, so that we convert our simulated reflectance from RGB to XYZ color space. We first measure the intensity of the substrate before printing the ink and the SRSs. After printing them, the intensity of the reflector is measured under the same lighting environment. Since our method assumes the reflectance of the substrate is one (which corresponds to the color white in the target images), we normalize the measured intensity of the reflector by the intensity of the substrate for comparison.

We fabricated two monotonic reflectors, each of which can display three colors, RGB and CMY, respectively. We chose these two reflectors since RGB and CMY are the vertices of the gamut represented by additive and subtractive color mixing. So, we consider that these reflectors correspond to cases difficult to reproduce. The three colors are displayed for three viewing directions, 0, 60, and 120 degrees. The elevation angle of the viewing directions is the same, 45 degrees. We obtained the six reflectances of the reflectors viewed from these three viewing directions. The averages of the absolute difference between the measured and the simulated reflectances for XYZ components are 0.01, 0.02, and 0.04. Our method faithfully predicts the reflectances of the real, fabricated reflectors. The relatively higher error in Z component is due to the fact that the UV printer we use is less accurate in printing colors with higher Z values. We confirmed this fact by printing a color chart.

Experiments on weights w_g and w_s . Fig. 7 demonstrates the effect of weights w_g and w_s in Eq (7), by setting w_g or w_s to zero. We omit the case $w_o = 0$ since in this case, E yields the minimum value zero when D_k are all zero, resulting in black images. Note that the images in this figure are created by rendering a virtual reflector by computer simulation. The input images are the same as those used for Fig. 11. Fig. 7(a) shows the results with $w_g = 0$. Our method then tries to maximize the intensity by making the height of the SRSs minimum, resulting in the obvious ghosting effects. Fig. 7(b) shows the results with $w_s = 0$ (i.e. the standard deviation term is ignored). The average intensities of the images differ considerably, resulting in an unnatural appearance when changing viewing direction. The ghosting effects are also visible. Fig. 7(c) shows the results without optimization of the SRSs. The parameters of the SRSs are randomly determined but the diffuse reflectance for each dotcell is optimized.

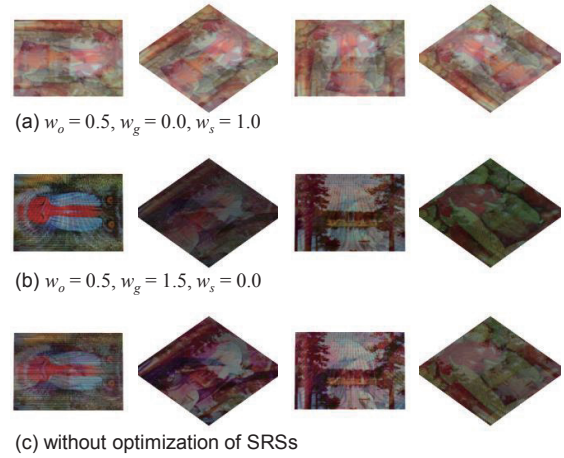


Fig. 7. Effect of weights w_g and w_s , and the effect of optimization.

Without the optimization, the ghosting effects are clearly visible. The optimized result can be seen in Fig. 11.

5.2 Evaluation of the contrast displayed

We evaluated the contrasts displayed by our reflectors. We conducted a set of experiments to reveal the relationships between the maximum contrast displayed by a single cell and the following three parameters:

- the azimuthal angle between two viewing directions
- the number of equally distributed viewing directions
- the elevation angle of the viewing direction

We focused on the maximum contrast displayed by our reflector from a theoretical point of view. That is, we did not take into account the physical constraints imposed by a real UV printer. In particular, we assumed the height of the SRSs to be 1.0mm and that no color mixing between adjacent dotcells occurred. These assumptions are not applicable to any currently available UV printer; the maximum height of the curable ink is usually less than 1.0mm and the ink between neighboring dotcells is easily mixed. In addition to the relationships described above, we investigated the dependence of the contrast on the resolution of the dotcell by measuring the radiances for dotcells with three different resolutions: 50×50 , 100×100 , and 150×150 .

In order to measure the maximum contrast, we used white (1.0) and black (0.0) images as input images. We measured the radiances from a single cell towards two adjacent viewing directions and computed the contrast as the ratio between the higher radiance and the lower radiance. More specifically, the contrast K was computed using the formula $K = (L_w - L_b)/L_w$, where L_w and L_b are the radiances for the directions corresponding to the white and black images, respectively. K ranges from zero to one; where zero signifies that there is no contrast. Figs. 8, 9, and 10 show the results of our experiments, showing the relationships between the maximum contrast and the above three parameters, respectively. We describe the details of the individual experiments below.

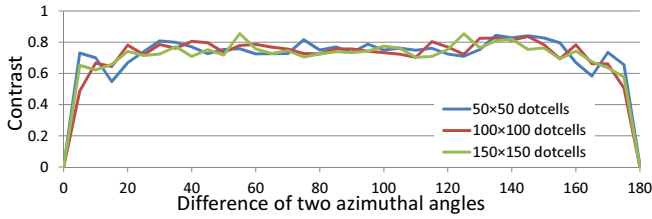


Fig. 8. Relationship between the contrast and the difference in azimuthal angle between two viewing directions. The vertical and horizontal axes indicate the contrast and the difference in degrees, respectively.

The measured radiances are actually different from the expected ideal radiances. We computed Eq. (3), the error function indicating the degree of difference for all the experiments shown in this section. The average error was 0.097. One common finding we obtained from these experiments is that the resolution of a cell has little effect on the contrast. This is observed in the figures; the three plots corresponding to dotcells with the three different resolutions are all similar (see Figs. 8, 9, and 10).

Dependence on the difference in azimuthal angle between two viewing directions: Fig. 8 was created by measuring the radiances for two viewing directions where the azimuthal angle of one viewing direction is fixed at zero degrees and the other one is varied from 0 to 180 degrees. Note that the SRSs in the two subcells corresponding to the two viewing directions become parallel when the difference in azimuthal angle between the two viewing directions is 0 or 180 degrees. In this case, the same color is displayed, resulting in no contrast. Fig. 8 indicates that our reflector can display a certain amount of contrast even if the difference is only five degrees. The contrast is relatively higher and almost constant when the azimuthal angle difference is between 20 and 160 degrees. However, there is some perturbation in the contrast, which is caused by the fact that the optimal SRS changes discretely with respect to the azimuthal angle.

Dependence on the number of equally distributed viewing directions: In this experiment, we changed the number of viewing directions whose azimuthal angles are sampled at a regular interval. The input images are one white image and N_b black images. The contrast is measured by the difference between the reflected radiance corresponding to the white image and that to the adjacent black image. Fig. 9 shows the relationship between the contrast and the number of black images N_b . The contrast slightly decreases in proportion to the number of input images.

Dependence on the elevation angle of the viewing directions: Fig. 10 was created by using the two target images and by varying the elevation angle of the corresponding two viewing directions from 5 to 90 degrees. The azimuthal angles are fixed to 0 and 90 degrees. Fig. 10 shows the contrast for the different elevation angles. All the colors are mixed when the viewing direction is normal to the surface, i.e., the elevation angle is 90 degrees. Thus, there is no

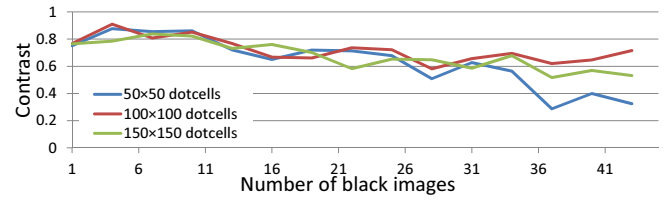


Fig. 9. Relationship between the contrast and the number of images. The vertical and horizontal axes indicate the contrast and the number of black images N_b , respectively.

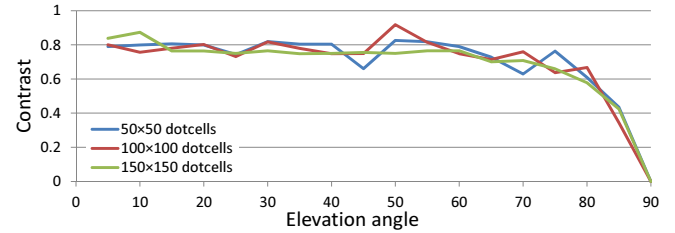


Fig. 10. Relationship between the contrast and the elevation angle of the viewing direction. The vertical and horizontal axes indicate the contrast and the elevation angle, respectively.

contrast. While a certain amount of contrast is observed for elevation angles up to approximately 70 degrees, the contrast decreases considerably for elevation angles larger than 70 degrees.

5.3 Practical Applications

This section show several applications of our reflector. The following reflectors require no special devices nor electricity. The user can observe the images displayed by the reflectors under the natural illumination or a standard indoor illumination condition.

The first example shown in Fig. 1 demonstrates the ability of our reflector to be fabricated on different material and/or objects. In this example, a set of reflectors displaying the SIGGRAPH logo are fabricated and we use them for decorating personal objects. For smartphone cases, the reflectors are directly fabricated on them by the UV printer. For other objects, the reflectors are fabricated on stickers and we paste them onto the objects. The reflectors can be put even on curved surfaces such as the pen case or duck toys. These reflectors exhibit different color variations of the logo depending on the viewing direction.

Next, in Fig. 11, we use our reflector for an artistic purpose; we call it *mirror art*. We fabricated the reflector using four input images shown in the inset of Fig. 11(b). Fig. 11(a) shows four reflected images of our reflector in the four mirrors placed around the reflector. In spite of the fact that all the mirrors reflect the same object (our reflector), we see a different image in each mirror.

In Fig. 12, unlike the other examples, we fix the viewing direction and change the direction of light illuminating the reflector. The reflector can then display different images depending on the lighting direction. We rendered six images of a teapot illuminated from six directions, respectively, and use them as the input images to our method. The fabricated reflector displays one of the images

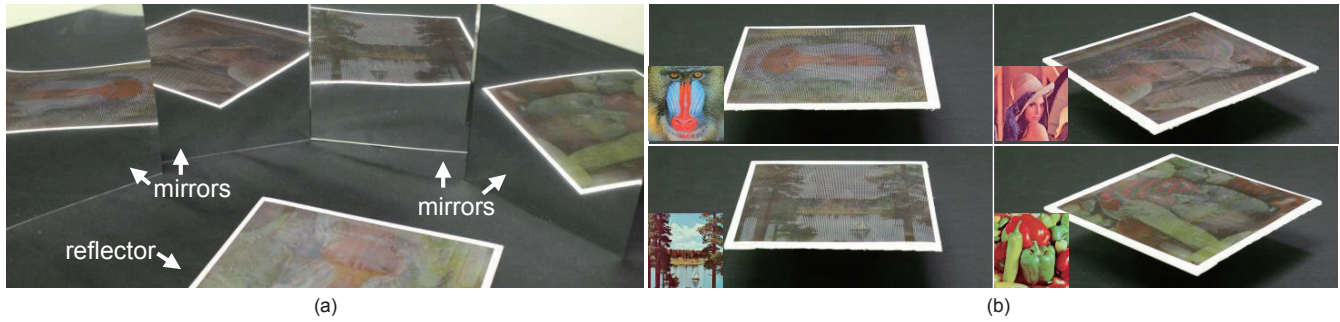


Fig. 11. Mirror art. (a) By placing mirrors around our reflector and viewing them from an appropriate position, the user sees different images in the mirrors even though there is only a single sheet on which our reflector is printed. (b) shows the same reflector viewed from four different directions. The input images are shown in the insets.

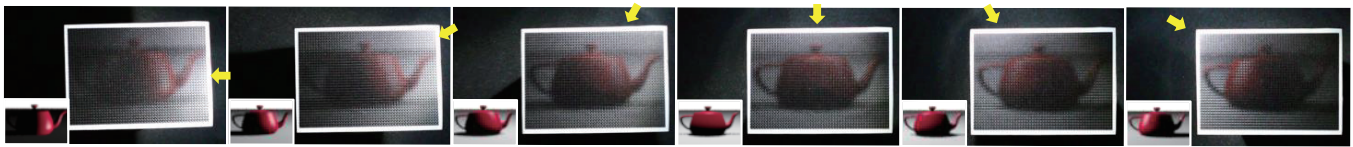


Fig. 12. Our reflector is used to display images of a teapot illuminated by the real light. The direction of light is indicated by the yellow arrows. The shading and shadows of the teapot change depending on the lighting direction as if there was a 3D teapot in the reflector. We use six input images for this example shown in the insets.

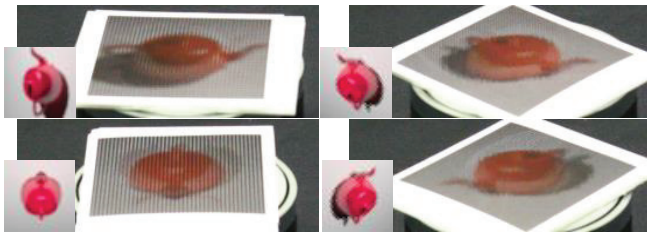


Fig. 13. Holographic reflector. By designing our reflector with the images of a teapot rendered from different view points, appropriate image is displayed according to the viewing direction as if there was a 3D teapot in the reflector.

according to the direction of real light illuminating the reflector. The user can learn how the shading and shadows change by altering the direction of real light by him/herself.

Finally, Fig. 13 demonstrates the use of our reflector for displaying a 3D object, *holographic reflector*. We prepared four input images of the teapot rendered from four different viewing directions (see the insets in the figure). The reflector can display the appropriate image from them according to the viewing direction. The user can see the 3D teapot through the reflector. Note that the shading of the teapot changes as well depending on the viewing direction.

Our method designs the SVBRDF only for N discrete viewing directions, and thus the reflectances for other directions are not defined. However, as shown in the accompanying video, our designed SVBRDF does not produce weird appearances for other viewpoints.

The proposed objective function in Eq. (7) does not consider the durability of SRSs. Therefore, short line segments can peel off from

the reflector. In our experiments, the lengths of the peeled line segments are less than 0.85mm, and these are isolated (i.e. they do not touch other line segments).

Currently, the number of displayable images in our method is limited to six mainly due to the resolution of the UV printer and the huge computation time required to optimize Eq. (7) for $N > 6$. We would like to optimize $E(G)$ efficiently to increase the number of displayable images, but this is left for future work.

One limitation of our method is that the images displayed tend to be darker than the original images. This is because they are displayed such that the SRSs subtract some portion of the reflected light. A simple solution to this problem is making the cell size larger so that each cell can receive a larger amount of light. However, since the maximum height of the SRSs is limited, the elevation angle of the viewing direction becomes lower, making it difficult to view the target images. This is a trade-off between the intensity and the visibility of the images. Another possible solution would be to use backlight behind the reflector. Combining a reflector printed on a transparent sheet with a backlight would address this problem.

6 CONCLUSION AND FUTURE WORK

We have proposed a method for fabricating a reflector that can display multiple color images. Our method designs a SVBRDF that reflects the incident light in multiple directions, which makes it possible to increase the number of displayable images compared to state-of-the-art methods. These complex reflections are reproduced by printing an ink layer and SRSs using a standard UV printer. The SRSs are optimized to minimize ghosting effects for better appearance. The color of the ink layer is calculated to match the

reflected light from the reflector with the target images, taking into account the gamut of the UV printer through the color reproduction process. We demonstrated that our simulation system can predict the appearance of the reflector faithfully and the fabricated reflector can display multiple color images without ghosting effects.

In the future, we would like to get faster computation of the microstructural stripe patterns and the color pattern of the ink layer by precomputation. The most time-consuming part is the computation of the color pattern of the ink layer, whereas computation of the SRS is rather efficient. We would like to precompute a set of color patterns for possible combinations of pixel colors in the input images. Moreover, to obtain better contrast of the displayed images, the SRS could be adapted to the content of the desired images; we can use a different SRS for each pixel. However, we need to construct the SRSs taking into account the coherence of the reflected radiances from neighboring cells. Otherwise, the appearance would be noisy. Moreover, we would like to incorporate durability of the SRSs into the objective function.

ACKNOWLEDGEMENTS

The authors would like to thank Pierre Poulin for proofreading of the first draft. The authors would also like to thank the reviewers for their constructive comments to improve our paper. This work was supported by JSPS KAKENHI Grant Numbers JP15H05924.

REFERENCES

- Marc Alexa and Wojciech Matusik. 2010. Reliefs As Images. In *ACM SIGGRAPH 2010 Papers (SIGGRAPH '10)*. ACM, ACM, New York, NY, USA, Article 60, 7 pages. <https://doi.org/10.1145/1833349.1778797>
- Ilya Baran, Philipp Keller, Derek Bradley, Stelian Coros, Wojciech Jarosz, Derek Nowrouzezahrai, and Markus Gross. 2012. Manufacturing Layered Attenuators for Multiple Prescribed Shadow Images. *Computer Graphics Forum (Proceedings of Eurographics 2012)* 31, 2 (May 2012), 603–610.
- Peter N. Belhumeur, David J. Kriegman, and Alan L. Yuille. 1999. The Bas-Relief Ambiguity. *International Journal of Computer Vision* 35, 1 (1999), 33–44. <https://doi.org/10.1023/A:1008154927611>
- Amit Bermano, Ilya Baran, Marc Alexa, and Wojciech Matusik. 2012. Shadowpix: Multiple Images from Self Shadowing. *Comput. Graph. Forum (Proc. EUROGRAPHICS 2012)* 31, 2 (2012), 593–602.
- Matthew Brand. 2011. Specular holography. *Appl. Opt.* 50, 25 (Sep 2011), 5042–5046. <https://doi.org/10.1364/AO.50.005042>
- Daniel Glasner, Todd Zickler, and Anat Levin. 2014. A Reflectance Display. *ACM Trans. Graph.* 33, 4, Article 61 (July 2014), 12 pages. <https://doi.org/10.1145/2601097.2601140>
- David E. Goldberg. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Professional.
- Roger David Hersch and Sylvain Chosson. 2004. Band Moiré Images. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 239–247. <https://doi.org/10.1145/1015706.1015709>
- Yanxiang Lan, Yue Dong, Fabio Pellacini, and Xin Tong. 2013. Bi-scale Appearance Fabrication. *ACM Trans. Graph.* 32, 4, Article 145 (July 2013), 12 pages. <https://doi.org/10.1145/2461912.2461989>
- Douglas Lanman, Matthew Hirsch, Yunhee Kim, and Ramesh Raskar. 2010. Content-adaptive Parallax Barriers: Optimizing Dual-layer 3D Displays Using Low-rank Light Field Factorization. *ACM Trans. Graph.* 29, 6, Article 163 (Dec. 2010), 10 pages. <https://doi.org/10.1145/1882261.1866164>
- Douglas Lanman, Gordon Wetzstein, Matthew Hirsch, Wolfgang Heidrich, and Ramesh Raskar. 2011. Polarization Fields: Dynamic Light Field Display Using Multi-layer LCDs. *ACM Trans. Graph.* 30, 6, Article 186 (Dec. 2011), 10 pages. <https://doi.org/10.1145/2070781.2024220>
- Anat Levin, Daniel Glasner, Ying Xiong, Frédo Durand, William Freeman, Wojciech Matusik, and Todd Zickler. 2013. Fabricating BRDFs at High Spatial Resolution Using Wave Optics. *ACM Trans. Graph.* 32, 4, Article 144 (July 2013), 14 pages. <https://doi.org/10.1145/2461912.2461981>
- Wojciech Matusik, Boris Ajdin, Jinwei Gu, Jason Lawrence, Hendrik P. A. Lensch, Fabio Pellacini, and Szymon Rusinkiewicz. 2009. Printing Spatially-varying Reflectance. *ACM Trans. Graph.* 28, 5, Article 128 (Dec. 2009), 9 pages. <https://doi.org/10.1145/1618452.1618474>
- Niloy J. Mitra and Mark Pauly. 2009. Shadow Art. *ACM Trans. Graph.* 28, 5, Article 156 (Dec. 2009), 7 pages. <https://doi.org/10.1145/1618452.1618502>
- Marios Papas, Thomas Houit, Derek Nowrouzezahrai, Markus Gross, and Wojciech Jarosz. 2012. The Magic Lens: Refractive Steganography. *ACM Trans. Graph.* 31, 6, Article 186 (Nov. 2012), 10 pages. <https://doi.org/10.1145/2366145.2366205>
- Alberto Pascual-Montano, J. M. Carazo, Kieko Kochi, and Roberto D. Pascual-Marqui. 2006. Nonsmooth Nonnegative Matrix Factorization (nsNMF). *IEEE Trans. on Pattern Analysis and Machine Intelligence* 28, 3 (March 2006), 403–415.
- Ken Perlin, Salvatore Paxia, and Joel S. Kollin. 2000. An Autostereoscopic Display. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 319–326. <https://doi.org/10.1145/344779.344933>
- Petar Pjanic and Roger D. Hersch. 2015a. Color Changing Effects with Anisotropic Halftone Prints on Metal. *ACM Trans. Graph.* 34, 6, Article 167 (Oct. 2015), 12 pages. <https://doi.org/10.1145/2816795.2818083>
- Petar Pjanic and Roger D. Hersch. 2015b. Color Imaging and Pattern Hiding on a Metallic Substrate. *ACM Trans. Graph.* 34, 4, Article 130 (July 2015), 10 pages. <https://doi.org/10.1145/2766944>
- Christian Regg, Szymon Rusinkiewicz, Wojciech Matusik, and Markus Gross. 2010. Computational Highlight Holography. *ACM Trans. Graph.* 29, 6, Article 170 (Dec. 2010), 12 pages. <https://doi.org/10.1145/1882261.1866192>
- Kaisei Sakurai, Dobashi Yoshinori, and Nishita Tomoyuki. 2017. A Method for Fabricating Reflectors Presenting Multiple Colored Images by Raised Linear Stripes. In *Proceedings of the The 5th IEEE International Workshop on Image Electronics and Visual Computing*. The Institute of Image Electronics Engineers of Japan, The Institute of Image Electronics Engineers of Japan, Da Nang, Vietnam, 6.
- Christian Schüller, Daniele Panofzo, and Olga Sorkine-Hornung. 2014. Appearance-mimicking Surfaces. *ACM Trans. Graph.* 33, 6, Article 216 (Nov. 2014), 10 pages. <https://doi.org/10.1145/2661229.2661267>
- Xavier Snelgrove, Thiago Pereira, Wojciech Matusik, and Marc Alexa. 2013. Parallax Walls: Light fields from occlusion on height fields. *Computers & Graphics* 37, 8 (2013), 974 – 982. <https://doi.org/10.1016/j.cag.2013.07.002>
- Wenhao Song, Alexander Belyaev, and Hans-Peter Seidel. 2007. Automatic Generation of Bas-reliefs from 3D Shapes. In *Proceedings of the IEEE International Conference on Shape Modeling and Applications 2007 (SMI '07)*. IEEE Computer Society, IEEE Computer Society, Washington, DC, USA, 211–214. <https://doi.org/10.1109/SMI.2007.9>
- Daniel Šýkora, Ladislav Kavan, Martin Čadík, Ondřej Jamříška, Alec Jacobson, Brian Whited, Maryann Simmons, and Olga Sorkine-Hornung. 2014. Ink-and-ray: Bas-relief Meshes for Adding Global Illumination Effects to Hand-drawn Characters. *ACM Trans. Graph.* 33, 2, Article 16 (April 2014), 15 pages. <https://doi.org/10.1145/2591011>
- Gordon Wetzstein, Douglas Lanman, Wolfgang Heidrich, and Ramesh Raskar. 2011. Layered 3D: Tomographic Image Synthesis for Attenuation-based Light Field and High Dynamic Range Displays. *ACM Trans. Graph.* 30, 4, Article 95 (July 2011), 12 pages. <https://doi.org/10.1145/2010324.1964990>
- Gordon Wetzstein, Douglas Lanman, Matthew Hirsch, and Ramesh Raskar. 2012. Tensor Displays: Compressive Light Field Synthesis Using Multilayer Displays with Directional Backlighting. *ACM Trans. Graph.* 31, 4, Article 80 (July 2012), 11 pages. <https://doi.org/10.1145/2185520.2185576>
- Tim Weyrich, Jia Deng, Connelly Barnes, Szymon Rusinkiewicz, and Adam Finkelstein. 2007. Digital Bas-relief from 3D Scenes. *ACM Trans. Graph.* 26, 3, Article 32 (July 2007), 32:1–32:7 pages. <https://doi.org/10.1145/1276377.1276417>
- Yonghao Yue, Kei Iwasaki, Bing-Yu Chen, Yoshinori Dobashi, and Tomoyuki Nishita. 2012. Pixel Art with Refracted Light by Rearrangeable Sticks. *Computer Graphics Forum* 31, 2pt3 (2012), 575–582. <https://doi.org/10.1111/j.1467-8659.2012.03036.x>