SHADOWPIX: Multiple Images from Self Shadowing

Amit Bermano[1,2] Ilya Baran[1] Marc Alexa[3,5] Wojciech Matusk[1,4]

[1]Disney Research Zürich [2]ETH Zürich [3]TU Berlin [4]MIT CSAIL [5]Disney Research Boston

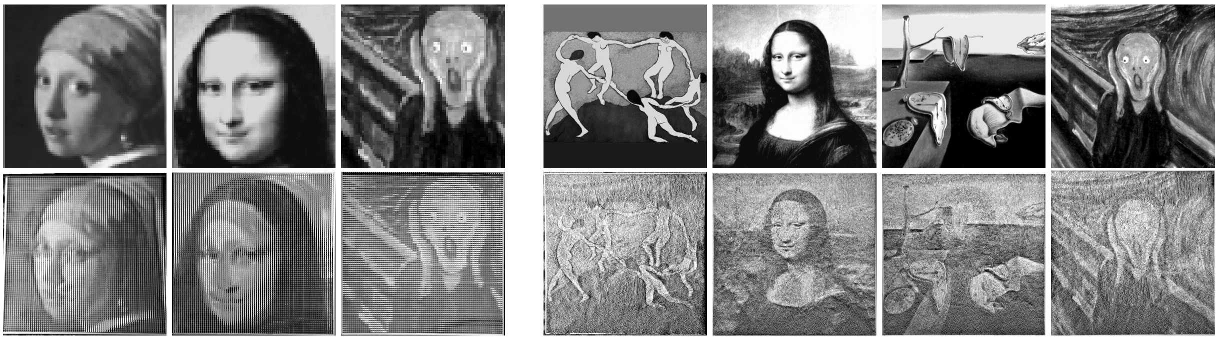

Figure 1: SHADOWPIX *are diffuse white surfaces whose self-shadowing pattern appears in the form of multiple prescribed images, depending on the illumination direction. The top row has seven input images. The bottom row has photographs of two* SHADOWPIX*, lit from different sides. We present two methods for constructing* SHADOWPIX*. The left three photographs are of a surface constructed with the* local *method, and the right four photographs are of a surface constructed with the* global *method. Due to poor downsampling in common software, we recommend viewing these images zoomed in to avoid aliasing.*

Abstract

SHADOWPIX *are white surfaces that display several prescribed images formed by the self-shadowing of the surface when lit from certain directions. The effect is surprising and not commonly seen in the real world. We present algorithms for constructing* SHADOWPIX *that allow up to four images to be embedded in a single surface.* SHADOWPIX *can produce a variety of unusual effects depending on the embedded images: moving the light can animate or relight the object in the image, or three colored lights may be used to produce a single colored image.* SHADOWPIX *are easy to manufacture using a 3D printer and we present photographs, videos, and renderings demonstrating these effects.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Shadowing

1. Introduction

The effect of illuminating a white plastic surface from different directions is usually easily predictable. In this paper, we show how to generate surfaces for which this is not the case: depending on the light direction, the shadows the surface casts on itself form different prescribed images. We call these surfaces SHADOWPIX. While our work is primarily an exploration of the space of intelligent material structures, SHADOWPIX are a step towards passive dynamic displays that interact with lighting (in the style of [NBB04] and [FRSL08]). Other potential applications include architectural design: for example, they can be used as walls that look different depending on the time of day, or as a sophisticated sundial. SHADOWPIX can be used in a theater setting, where precisely controlled directional illumination is readily available. Another possibility is a steganography-like application, where an object reveals a secret image only under specific illumination. Finally, ShadowPix can simply be fascinating memorabilia.

To concentrate on the self-shadowing effect, we consider ShadowPix to be made of a diffuse, uniformly white material, with minimal subsurface scattering. We assume the surface to be a grid of elements with square bases and varying heights (Figure 8, top), and later slightly relax this assumption (Figure 6). The top of each element is planar and parallel to the ground plane and we assume that each element is either fully lit or fully shadowed for a certain light direction. Thus, the images generated by the surface through self-shadowing for a certain light direction are binary. Consequently, grayscale images can be reproduced only by halftoning.

Since we need to halftone the input image, we necessarily have the competing goals of similarity to the input image and supporting high-resolution images. In addition, we assume the viewer sees all elements equally well. This is true for an idealized orthographic viewer or, more practically, if the height variation is small. This additional requirement of low height variation also makes ShadowPix easier and cheaper to manufacture because tall thin rigid spikes are fragile.

While there are potentially more general models for ShadowPix, discrete height-fields have the advantage of being easy to manufacture. The resulting objects can be constructed with a wide range of processes; while we created the prototypes with 3D printing, milling or creating molds would also be easy. Similar models have been used for a long time outside the computer graphics field, like the pinscreen, which was invented by Alexandre Alexeieff, Claire Parker, and Alexandra Grinevskaya in the 1930's and used to produce several short films using the self-shadowing effect.

For producing ShadowPix, we have developed a local and a global algorithm. The algorithms roughly correspond to similar halftoning strategies. For the local algorithm, a gray-value image pixel is represented by a small square of surface elements. We separate the elements inside each square into shadow-casters and receivers a priori. A certain percentage of the square will be in shadow, corresponding to a grayscale value in the image. This scenario yields an explicit construction algorithm. However, at most three images can be produced this way and they are not ideally exploiting the available printer resolution.

In the global scenario, every element can both cast and receive shadows, allowing us to produce four images and use the full printer resolution by having a printer element represent an image pixel. The complexity of the objective function required us to use stochastic global optimization techniques over a reasonable subset of all possible height vectors. Figure 1 shows ShadowPix produced by both methods.

We present manufactured results for both algorithms, showing that our techniques work in practice. We demonstrate that the encoded images can be unrelated, they can be frames of a video, lighting configurations, or color channels of a single image.

2. Related Work

Computing shape from shadows (darkness) is a classic problem studied by computer vision researchers. In this problem, a 3D surface is reconstructed from a set of shadow images. The initial work of Shafer and Kanade [SK83] describes geometrical constraints imposed by shadows. The first algorithms [KS86, RPL89] assume that the scene is a height field (terrain), the camera is fixed and orthographic, and the light is directional and is moving along a known arc. These algorithms use a shadowgram data structure to encode shadow dependencies, but they typically reconstruct only a single cross-section of the surface. Daum and Dudek [DD98] extend the shadowgram algorithms to compute a complete 2D height field. Yu and Chuang [YC02] propose shadow graphs, a more efficient representation for encoding constraints in a terrain surface. The general setting for the problem of shape from shadows appears to be similar to ours at a high level. However, there are key differences. First, our input images are grayscale and not binary, as typical shadow images. Furthermore, our goal is different: we do not attempt to reconstruct the actual 3D scene geometry but only produce a height field that induces the desired images. Finally, we also try to keep the height of the terrain as small as possible in order to reduce view-dependent effects and to make the fabrication process more robust.

Our work is also related to shadow sculptures—3D structures that can generate up to three desired shadow images on planar surfaces when illuminated from different directions. These shadow images are effectively silhouettes of the shadow sculpture from different views. Shadow sculptures have been typically assembled by trial and error by artists (e.g., Tim Nobel and Sue Webster, Shigeo Fukuda). Elber [Elb02] has constructed sculptures with prescribed views from different directions. More recently, computational methods [KvWW09, MP09] have considered automatically constructing shadow sculptures. Kieren et al. [KvWW09] give algorithms to determine whether a connected 3D shape can have 3 given rectilinear silhouettes, but provide no hint for how to adjust the silhouettes to be attainable by a 3D object. Shadow Art [MP09] partially addresses this problem, showing how to continuously warp silhouettes to "match" in 3D. Shadow sculptures display shadows on external planar surfaces, whereas a ShadowPix serves as both the shadow caster and the shadow reciever (hence, self-shadowing). The main advantages of using self-shadowing are simplicity of manufacturing (ShadowPix are just uniformly white heightfields) and the ability to display halftoned images using a single connected object: Mitra and Pauly [MP09] demonstrate an example with images, but

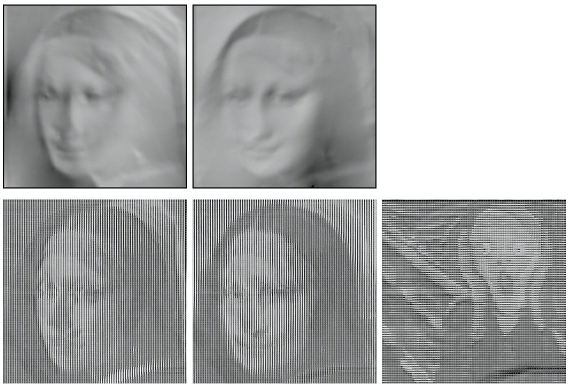

Figure 2: *Physically-based renderings of a relief generated by Alexa and Matusik's method [2010] from the first two input images in Figure 1 (top) and our local method (bottom) under ideal viewing and lighting conditions. Our methods capture more detail and attain significantly higher quality, while simultaneously embedding more images.*

it requires embedding the sculpture into a transparent cube and only a simulated result with perfect lighting is shown.

The most relevant prior work [AM10] constructs relief surfaces that can form two images for different light directions. This method only uses diffuse shading and assumes no self-shadowing or other shading effects. Focusing only on shading makes the problem simpler due to its local nature, but it results in significant limitations. In particular, creating more than two images using only shading is impossible since normals have only two degrees of freedom. Even with two images, the requirement that the normal field be integrable is an additional constraint, which results in severe ghosting artifacts between the two images (Figure 2). In contrast, shading due to self-shadowing is not a local effect: a surface point can cast a shadow on another point that is potentially far away. This non-locality makes it more challenging to control self-shadowing, but in return, it allows us to encode more images into one surface and obtain sharper results.

Our global method is structurally similar to some optimization-based halftoning methods [AA92, PQW*08]. These halftoning methods maximize the similarity to a single given image and can directly set the pixels of the binary image. In contrast, we are maximizing similarity to four images simultaneously, we can only influence the binary images by changing heightfield element heights, and have additional manufacturing considerations, such as height and surface smoothness.

3. Method

We offer two different methods for generating ShadowPix. The local method uses a grid corresponding to the pixels of the input images and distinguishes between shadow receivers, making up most of each pixel, and walls casting shadows. The local method can handle up to three different input images to be displayed for three light directions, which are aligned with the grid of elements, when projected to the xy plane.

The global method individually assigns height values to the smallest printable elements. Each element is either shadowed or not. This binary pattern can be used to dither the input images. For this method we can potentially use an arbitrary number of input images as well as arbitrary light directions, but for reasons of efficiency in the optimization, it is beneficial if up to four light directions are considered, all being aligned with the grid of elements.

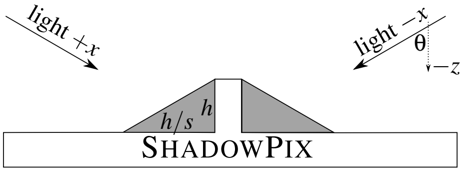

Figure 3: *A single height field element casting different shadows depending on the light direction.*

For both methods we use the following variables (Figure 3): the ground plane for the ShadowPix is parallel to the xy plane and the view direction is along the z-axis. We assume that all light sources are directional. For clarity of exposition, we assume that the light directions all form the same angle θ with the z-axis (i.e., have the same elevation over the ground plane). Then an element of height h casts a shadow of length $h/\cot\theta = h/s$ onto a plane parallel to xy. In the following discussion, we will make use of the proportionality constant s, which depends only on the setup of the lights.

3.1. Local Algorithm

The shadow receivers are arranged as a grid of unit squares to match the pixels of the images. The shadow casters are the walls between the squares. As shown in Figure 4, the heights of the receivers are $r_{i,j}$ and the heights of the x and y casters are $u_{i,j}$ and $v_{i,j}$, respectively. The three light directions are towards $+x$, $-x$, and $+y$. We admit that this makes the formulation asymmetric, but aligning the light directions with the pixel grid keeps the shadow geometry simple and leads to a very efficient algorithm.

Let the grayscale values of the images be $I^1_{i,j}$, $I^2_{i,j}$, and $I^3_{i,j}$. We need the unshadowed fraction of each square from each light direction to be the prescribed grayscale value. The constraints this requirement imposes on the element heights are:

$$\begin{aligned} u_{i,j} - r_{i,j} &= sI^1_{i,j} \\ u_{i+1,j} - r_{i,j} &= sI^2_{i,j} \\ v_{i,j} - r_{i,j} &= sI^3_{i,j}. \end{aligned} \tag{1}$$

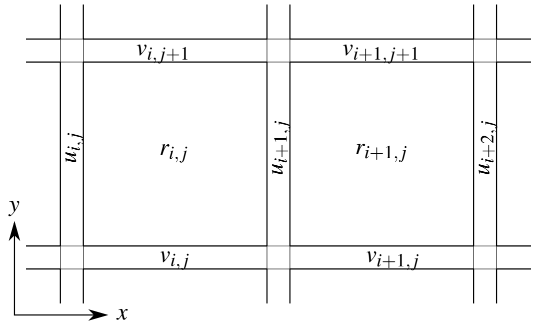

Figure 4: *A view from above of the grid and heights used by the local method.*

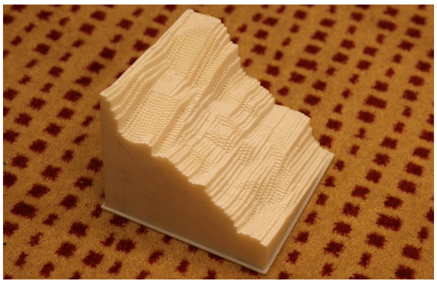

Figure 5: *Without chamfering, the local method produces* ShadowPix *that are too tall to be usable. The height increase is necessary to prevent a part of the surface closer to the* $+y$ *light direction from shadowing a part of the surface farther from the light direction (see Equation (3)). Chamfering produces* ShadowPix *whose maximum height is about 1 cm for an image that is 20 cm on each side.*

Note that this system of equations provides necessary conditions for the solution. In addition we require inequality constraints that make sure that only the walls are casting shadows and that these shadows are contained only in the pixels adjacent to the walls. The first condition simply means $r_{i,j} < \min(u_{i,j}, u_{i+1,j}, v_{i,j}, v_{i,j+1})$. With this condition in place, the latter requirement imposes a restriction on the height difference between adjacent walls. For example, in the $+x$ direction, if $u_{i,j} - u_{i+1,j} > s$ then the shadow cast by $u_{i,j}$ may spill over onto the pixel at $(i+1, j)$. For the$-x$ and $+y$ light directions, this leads to the requirements $|u_{i,j} - u_{i+1,j}| < s$ and $v_{i,j} - v_{i,j+1} < s$, respectively. We note that these restrictions do not limit the space of images we can represent.

A generic way to solve this problem would be to use linear programming. However, we note that the structure offers a particularly efficient assignment of heights in scan line order, i.e. the system is almost triangular.

First, we combine the first two linear equations to get a condition that determines heights along scan lines:

$$u_{i,j} = u_{0,j} + s \sum_{x=1}^{i} \left(I^2_{x,j} - I^1_{x,j}\right). \qquad (2)$$

We note that this directly satisfies the inequalities among the $u_{i,j}$. Thus we just walk along a line, starting at $u_{0,j}$ and determine the values of $u_{i,j}$. Now, the interesting observation is that the inequality $r_{i,j} < v_{i,j}$ can be rewritten in terms of the $u_{i,j}$ by exploiting their inequalities with $r_{i,j}$ as

$$u_{i,j+1} \geq u_{i,j} + s\left(-I^1_{i,j} + I^1_{i,j+1} - I^3_{i,j+1}\right), \qquad (3)$$

leading to a condition to be satisfied when moving to the next scan line. We can always satisfy this condition by making $u_{0,j+1}$ sufficiently larger than $u_{0,j}$. As we would like to minimize the height, we pick $u_{0,0} = 0$ and incrementally pick $u_{0,j}$ to be the smallest nonnegative value that still satisfies this inequality for all i. In this setup, no degrees of freedom are left for a fourth image.

This method can reproduce any three images for the idealized case of an orthographic viewer and unconstrained height differences. However, with the viewer at a finite distance and considering practical limitations of the construc-

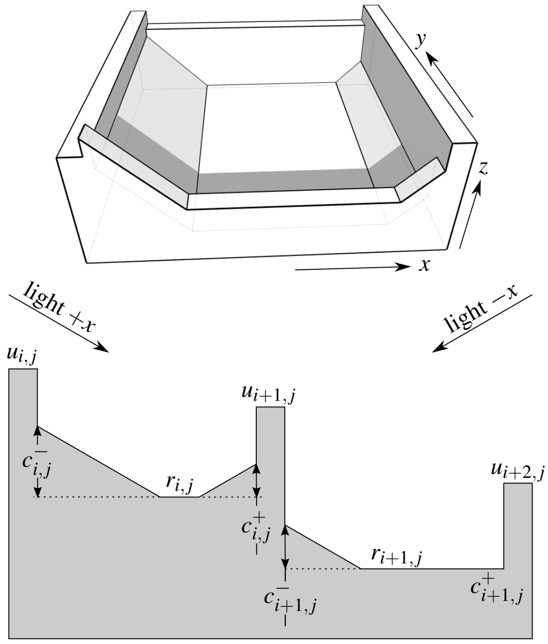

Figure 6: Top: *A single chamfered pixel produced by the local method.* Bottom: *The x-z cross-section of two chamfered pixels, illustrating the parameters. In this case, because there is no chamfer on the right of the second pixel,* $c^{+}_{i+1,j} = 0$. *Our algorithm would never actually produce this configuration because* $c^{-}_{i,j}$ *could be increased further, allowing* $u_{i+1,j}$ *and therefore* $r_{i+1,j}$ *to increase.*

tion process, we need to limit variation in height, because otherwise, the results are unusable (Figure 5). While it would be possible to add additional linear and non-linear constraints to limit the height variation, the number of degrees of freedom in the pixel model is too small to improve the height significantly. Instead, we address the problem of height variation by adjusting the geometry of the shadow receivers, which we have so far assumed to be flat. The simplest configuration we have found to be satisfactory works as follows: we add chamfers to the $+x$ and $-x$ edges of each shadow receiver, as shown in Figure 6. While these chamfers contradict our assumption that the surface is parallel to the ground plane, we have found that when they are produced with the 3D printer, their reflectance behaves as expected.

The chamfers are always parallel to the opposite light directions, so the chamfer at the $+x$ edge is never lit by the light towards $-x$ and vice versa. The heights of the chamfers at the walls, given by $c^-_{i,j}$ and $c^+_{i,j}$, are therefore additional degrees of freedom that affect the $+x$ and $-x$ images independently. The chamfers are potentially lit by their light directions, and are brighter than the base of the pixel because their normals are closer to the light directions. They may also shorten the shadows of their respective lights. Therefore, increasing $c^-_{i,j}$ always brightens the pixel (i,j) when lit by the $-x$ light and has no effect on the $+x$ light configuration. The brightness of the pixel when lit by the $-x$ light depends monotonically and piecewise-linearly on $c^-_{i,j}$. The situation is symmetric for $c^+_{i,j}$. We note that the piecewise linear function would be difficult to incorporate into a general linear system, but our incremental construction makes it easy to solve for appropriate values, as we explain below.

Suppose we computed row j using Equation (2), assuming $u_{0,j} = 0$. We then greedily introduce non-zero chamfer heights to adjust the r's to vary as little as possible. Specifically, we walk over the pixels in the row in order of increasing i and for each pixel, compare $r_{i,j}$ to $r_{i+1,j}$. If $r_{i+1,j}$ is higher, we increase $c^+_{i+1,j}$, which brightens the pixel $(i+1,j)$ lit by the $+x$ light, allowing us to lower $r_{i+1,j}$ and all u's and r's to the right of it and thus keep the illumination unchanged, while reducing height variation. We increase $c^+_{i+1,j}$ until either $r_{i+1,j} = r_{i,j}$ or $c^+_{i+1,j} = u_{i+2,j} - r_{i+1,j}$. If, on the other hand, $r_{i+1,j}$ is lower than $r_{i,j}$, we increase $c^-_{i,j}$ and raise $u_{i+1,j}$, $r_{i+1,j}$ and all of the r's and u's to their right, while keeping illumination unchanged. In this case, we stop when $r_{i+1,j} = r_{i,j}$, or when $c^-_{i,j} = u_{i,j} - r_{i,j}$, or when the chamfers collide and $c^-_{i,j} + c^+_{i,j} = s$.

The dot product of the $+y$ direction with the normal of a chamfer is smaller than with the normal of the pixel floor (the z direction), so chamfers make a pixel potentially less bright when lit by the $+y$ light. We compensate for this when computing the v's, but in some cases even when the pixel is completely unshadowed, it ends up insufficiently bright due to the chamfers. In this case, we uniformly reduce the brightness of I^3. In practice we have not had to reduce the brightness by more than 10%, and the effect is not noticeable.

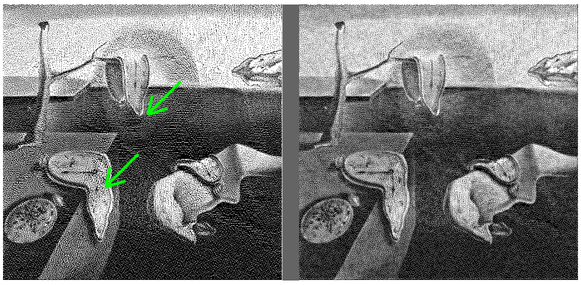

Figure 7: *A binary image produced by the global method without the gradient term (left) and with the gradient term (right). The four target images are those in Figure 1, right. The gradient term leads to better reproduction of thin features, like the clock hands and reduces the ghosting from Mona Lisa's mouth.*

This chamfering dramatically reduces the overall height of SHADOWPIX and improves quality. We have experimented with several variations of chamfering and they performed similarly.

3.2. Global Algorithm

For the global algorithm, we make no assumptions about the structure of the surface: each heightfield element can both cast and receive shadows. We assume that the light directions are aligned with the grid of elements. This simplifies the shadow computations, as each element can only shadow other elements along the grid directions. Moreover, we assume that the heights are multiples of s so that each element is either completely lit or completely in shadow. This allows us to treat the resulting images as binary. For the moment, we consider the four different images for light sources along the $\pm x$ and $\pm y$ directions. Our goal is to make these four binary images good dithered representations of the four input images while ensuring that the heightfield can be manufactured.

Objective Given a heightfield $h_{i,j}$, let $L^d_{i,j}(h)$ be the binary function indicating whether the element at (i,j) is lit when the surface is illuminated from direction d. Our goal is to find h such that $L^d_{i,j}(h)$ looks as close to $I^d_{i,j}$ as possible, for all d. This goal is similar to image halftoning, but we have more than one image and only indirect control over individual pixels. Many halftoning methods measure the difference between the target grayscale image and binary image by modeling the human visual system as a convolution with a low-pass filter p. This leads to the following objective:

$$\sum_{d=1}^{4} \|p * L^d(h) - p * I^d\|^2, \tag{4}$$

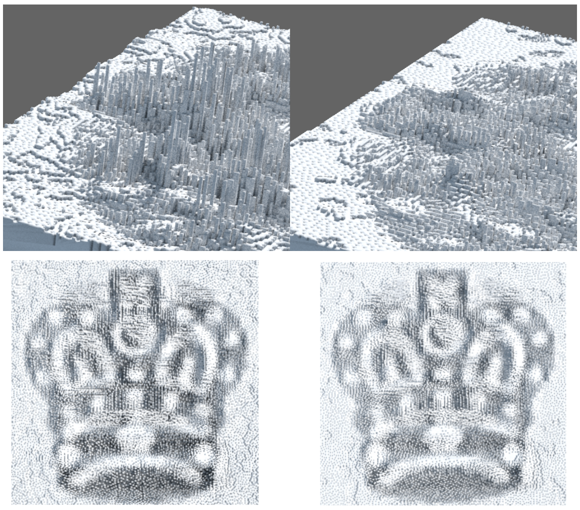

Figure 8: *Physically-based renderings of an image and the surface produced by the global method without the smoothing term (left) and with the smoothing term (right). Smoothing slightly reduces contrast, but significantly reduces the height variation.*

where $*$ denotes a convolution. Unlike in regular image halftoning, SHADOWPIX have lower contrast than the original image, so preserving edges is an important concern. We therefore do not convolve the target images with p, leading to increased sharpness. To further improve quality, we incorporate a term to match the gradients, and not just the absolute pixel values:

$$\sum_{d=1}^{4} \|G * p * L^d(h) - G * p * I^d\|^2, \tag{5}$$

where G is the finite difference gradient filter. Figure 7 illustrates the effect of this term. We have also experimented with other saliency maps, but found that using the gradient yields nearly identical results. An additional concern for manufacturing is that the heightfields should be as smooth as possible—otherwise, fragile heightfield elements may break off and low heightfield elements surrounded by high ones can be difficult to paint (Figure 8). To improve smoothness, we add a term that penalizes the gradient of the heightfield. Our final objective is:

$$\begin{aligned} &\sum_{d=1}^{4} \|p * L^d_{i,j}(h) - I^d_{i,j}\|^2 + \\ +w_G &\sum_{d=1}^{4} \|G * p * L^d_{i,j}(h) - G * p * I^d_{i,j}\|^2 + \\ +w_S &\|G * h\|^2, \end{aligned} \tag{6}$$

where w_G is the importance of the gradient term and w_S is the importance of the smoothness term. For all our experiments, we used the values $w_G = 1.5$ and $w_S = 0.001$, with image dimensions measured in pixels, intensities between 0 and 1, and the height measured in units of s. For the point spread function p, we simply use a 5-by-5 pixel Gaussian kernel; a more sophisticated perceptually-motivated p could be used instead.

Optimization Earlier optimization-based halftoning methods (for instance, [AA92]) typically used "direct binary search," which is essentially greedy objective descent. However, for more complicated objective functions [PQW*08] and for our problem, the greedy approach tends to get stuck in local minima. We therefore decided to use simulated annealing [KGJVM83] to optimize the objective. Our configuration is the heightfield $h_{i,j}$, which we initialize to all 0's. To transition between configurations, we change a single heightfield element up or down by up to $5s$. The transition is made if either the new objective function is lower or with probability $e^{(f_{t-1}-f_t)/T}$, where f_{t-1} and f_t are the previous and current objective function values and T is the current temperature. We restrict the maximum length of a shadow that's cast by one element in one image to 10. This amplifies the smoothness effect and helps create a binary pattern that is more random, which is a desirable feature when halftoning is concerned. For the 20cm-by-20cm SHADOWPIX, we use 10^9 steps and a linear cooling schedule. In our experiments, simulated annealing converged to a point whose objective was roughly half of the local minimum found by greedy search and the image fidelity was significantly higher (Figure 14).

This procedure is fairly time consuming, so we implemented a simple optimization to take advantage of multicore parallelism. Every height-field element of height hs, will cast a shadow whose length is at most h pixels. Because of this, each height change affects the blurred image only within a radius r, where r is the maximum of the height before and after the change (but at most 10) plus the radius of the blur kernel p. The effects of changing heightfield elements more than $2r$ apart are therefore guaranteed to be independent. Therefore, at every step, we pick multiple height-field elements at distances at least $2r$ from each other. We pick as many elements as we have cores and in parallel process the effect on the objective function of changing each of them.

4. Results and Evaluation

To test our construction methods, we designed and printed several SHADOWPIX, using both the local and global method. The methods were run on a 1.6GhZ Core i7 mobile CPU, the global method taking advantage of all eight cores. Our algorithms require a negligible amount of memory. To manufacture the SHADOWPIX we used an Objet Connex350 printer and an opaque black material (VeroBlack) with minimal subsurface scattering, which was then painted white with an airbrush. Based on the printer resolution, for the

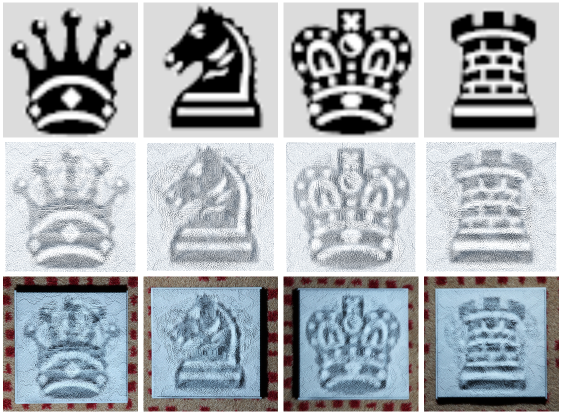

Figure 9: *Input images (top row), renderings (middle row), and photographs of a single* SHADOWPIX *produced by the global method (bottom row). This case is particularly challenging due to the extreme brightness differences required by the different black-and-white chess pieces.*

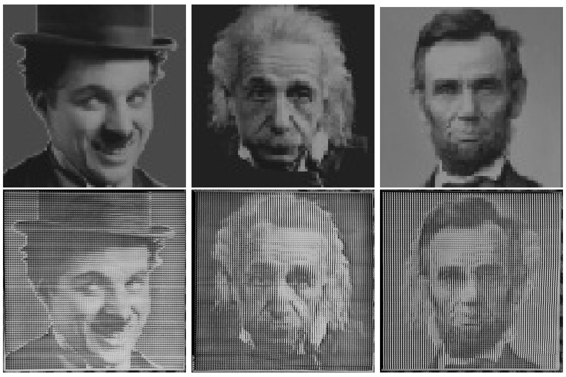

Figure 10: *Input images (top row) and photographs of a single* SHADOWPIX *produced by the local method (bottom row).*

local method we chose a pixel size of 2.5x2.5mm and a wall thickness of 0.25mm, which means that 81% of the total surface area is covered with receiving elements, and the shadowing of the other 19% is arbitrary. For the global method, we used an element size of 0.5x0.5mm because thinner heigtfield elements tend to bend or break. Our samples varied between 10 and 20 centimeters in each dimension.

We compared the printed results with renderings produced using a physically-based renderer; the viewing and light directions are slightly perturbed against the idealized assumptions to simulate the error in a real setup. Figure 9 shows that the rendering captures all of the qualitative features of the physical result, up to color balance. The ability to test with renderings considerably sped up the design process.

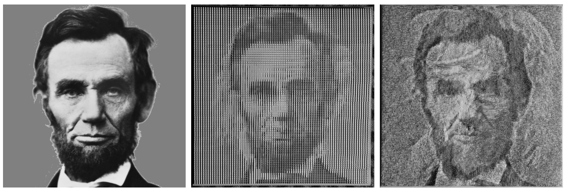

Figure 11: *An input image (left) and the image produced with the local method (middle) and the global method (right). The* SHADOWPIX *are the same size and were printed with the same 3D printer resolution. The global method image has higher resolution, higher contrast, and shares the* SHADOWPIX *with three other images (as compared to two for the local method), but it is noisier, has more ghosting, and is more sensitive to light and viewing directions.*

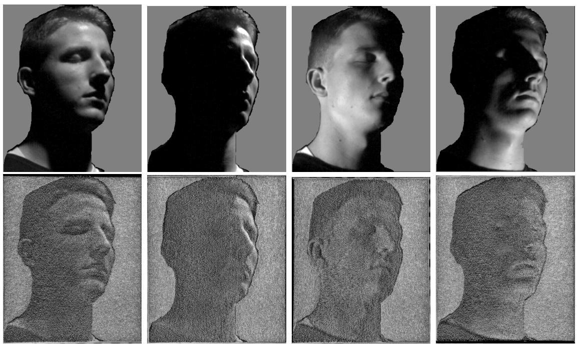

Figure 12: *Input images (top row) and photographs of a single* SHADOWPIX *produced by the global method (bottom row). When this* SHADOWPIX *is lit from a direction, the image appears as if lit from the opposite direction, for an unusual effect.*

We experimented with distinct portrait photographs (Figure 10) and paintings (Figure 1). We found that the local and global methods have their advantages and disadvantages. Figure 11 shows a side-by-side comparison of the same Lincoln's image produced by both methods. The local method produces a cleaner image, has less ghosting, while the global method is higher-resolution, has higher contrast, and can accomodate up to four images. We also compared their robustness with respect to deviations from ideal viewing and lighting conditions and found that the local method is significantly more tolerant (Figure 16). For many applications, this is an advantage, but the more startling effect of the image forming on a global SHADOWPIX may sometimes be preferred.

While the local method generates the surface nearly instantaneously (taking 6 seconds on our largest examples), the global method requires stochastic optimization and takes a couple of hours to run (ranging from from 29 minutes for the smaller chess pieces example, up to 104 minutes for

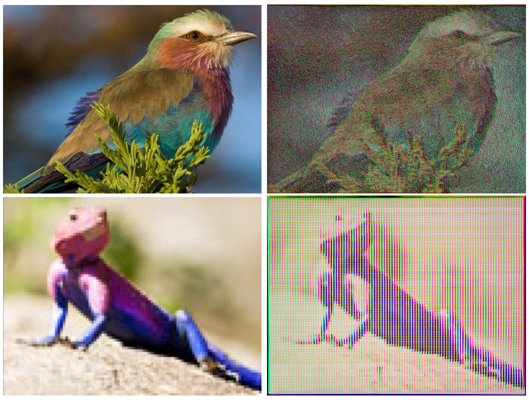

Figure 13: *Top Left: An input photograph of a bird. Top Right: A white SHADOWPIX that encodes the red, green, and blue channels of the input as separate images, illuminated by colored lights from three directions, using the global method. Bottom Left: A lower res input photograph of a lizard. Lower Right: A white SHADOWPIX that encodes the RGB channels, using the local method. These images were obtained by capturing three photographs in raw mode, each with a single color light, and adding them in software.*

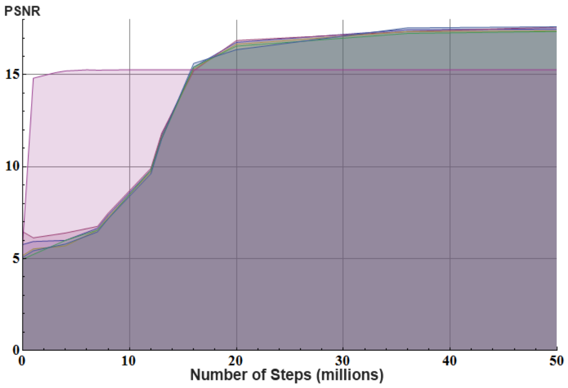

Figure 14: *This plot compares the greedy optimization convergence (magenta) to that of simulated annealing on the chess pieces example (other colors). Several simulated annealing runs are shown, demonstrating that the qualitative behavior is stable with respect to the stochastic choices. Not surprisingly, the greedy algorithm converges much faster, but it's solution is significantly (almost 3dB) worse than that found by simulated annealing.*

the larger paintings example) per SHADOWPIX. It took 2–6 hours to print each SHADOWPIX and about two hours for cleaning and painting, so we did not expend too much effort to get the fastest possible convergence.

We then experimented with using related images in SHADOWPIX. We achieved an unusual lighting effect by encoding four photographs of a face lit from different sides and associating them with the opposite light direction (Figure 12), so that when the SHADOWPIX is lit from a direction, it appears as if lit from the opposite one. We encoded the three color channels of a single image into a SHADOWPIX, so that when the image is lit by colored lights from different directions, the white surface displays a color image. This experiment was done both for the local and global method (Figure 13). Alexa and Matusik [AM10] have done a similar experiment, but their method is limited to two color channels, which is insufficient for most color images. We also encoded the animation frames of a running horse, so that when the light moves around the SHADOWPIX, an animation can be seen. Our video shows this and other results.

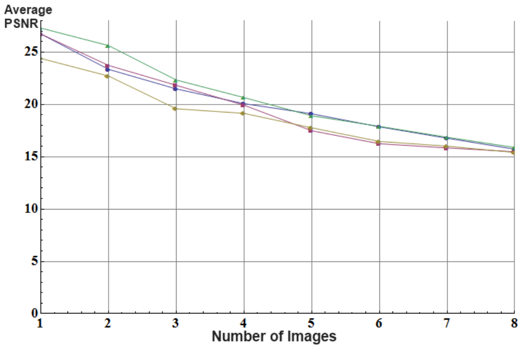

Figure 15: *This plot shows how the embedded image quality decreases as more images are embedded into a single SHADOWPIX using the global method. The different curves show four random embedding orders of the faces and the paintings.*

Quantitative error analysis We have performed several experiments to analyze the global method. For image quality evaluation, we used peak signal-to-noise ratio (PSNR) between the input image and the output bitmap convolved with the point spread function p. Figure 14 shows how simulated annealing converges on the chess pieces example and why we did not use greedy optimization. It also demonstrates that our result quality is largely independent of the initial condition and random choices made during a run.

We also looked at how encoding more images into a single SHADOWPIX affects the image quality. For this experiment, we added support for encoding images for the four diagonal light directions towards $(\pm x, \pm y)$; this involves only a change to the objective function: the shadows are calculated on four quarters of each pixel. Figure 15 shows the quality steadily decreasing as more images are added. While four images are not the absolute limit of the global method, embedding more images reduces the result quality below what we think is acceptable. The different curves in the figure

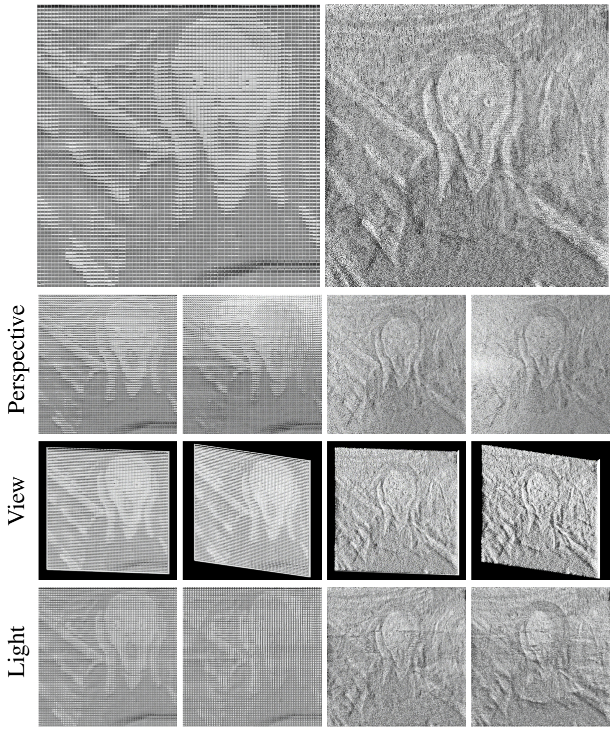

Figure 16: *We use physically-based rendering to explore how sensitive* ShadowPix *are to deviations from ideal viewing conditions, showing the local method on the left and the global method on the right. For the top images, we use an orthographic camera, orthogonally positioned, and a light at the correct angle. In the next row, the camera has been changed to have a field of view of 50 and 100 degrees. In the third row, we keep the camera orthographic, but change the view direction by 15 and 30 degrees. In the bottom row, we show the effect of changing the light direction by 20 and 40 degrees. These experiments show that the image produced by the local method is much more stable.*

show that adding images in different order has a significant influence on the effect, confirming that some images are more difficult to embed.

5. Conclusions

In this paper, we have explored the use of surface self-shadowing as a means of embedding multiple images into a surface when it is lit from different directions. We have shown two different methods that accomplish this task. The first method, the local method, uses geometric primitives that consist of thin walls placed on a rectangular grid. Each primitive represents a light-dependent pixel that can reproduce different grayscale values when lit from three different directions. We have shown a deterministic algorithm that allows building a surface from these primitives while minimizing the overall surface height. Our second method, the global method, represents a surface as an arbitrary height-field. Our key contribution for this method is an appropriate objective function that preserves perceptually important image features and keeps the surface height small. This objective function is optimized using simulated annealing to produce height fields that embed up to four different grayscale images into a single height field. We have evaluated both methods by producing many different surfaces that generate multiple images when lit from different directions.

The problem presented in the paper is new and therefore there are many opportunities for future work in this research area. In particular, in the context of the local method, we would like to explore different geometric primitives, for example, primitives based on a hexagonal grid. For the global method, it would be desired to have a deterministic algorithm that is fast and computes surfaces of a quality similar to that of our optimization-based approach. In our methods we have not accounted for the effects due to the global illumination and we suspect that interreflections between parts of the white surface are responsible for some of the ghosting and contrast reduction. We believe that accounting for these effects would improve the quality of the resulting images. At least in the case of the local method, indirect illumination effects could be precomputed for each primitive configuration. We have only used rapid prototyping methods to manufacture our surfaces, but we would like to explore different manufacturing methods including injection molding and CNC milling.

Looking into the future, we believe that it would be desirable to exploit both geometry (e.g., due to self-shadowing and surface normals) and reflectance (e.g., diffuse, glossy, and specular effects) in order to obtain additional degrees of freedom and encode even more images in a single surface. Furthermore, extending our methods to non-planar surfaces can broaden the range of possible applications. Finally, one can envision building a dynamic height field display that would exploit the concepts described in this paper and allow changing the displayed images over time.

Acknowledgements

We thank Bernd Bickel for help with the 3D printer and Christian Regg for help with painting.

References

[AA92] Analoui M., Allebach J.: Model-based halftoning using direct binary search. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* (1992), vol. 1666, pp. 96–108. 3, 6

[AM10] Alexa M., Matusik W.: Reliefs as images. *ACM Transactions on Graphics 29*, 4 (July 2010), 60:1–60:7. 3, 8

[DD98] Daum M., Dudek G.: On 3-d surface reconstruction using shape from shadows. In *IEEE Computer Vision and Pattern Recognition* (1998), pp. 461–468. 2

[Elb02] Elber G.: "Beyond Escher for real" project, 2002. 2

[FRSL08] Fuchs M., Raskar R., Seidel H.-P., Lensch H. P. A.: Towards passive 6d reflectance field displays. *ACM Transactions on Graphics 27*, 3 (Aug. 2008), 58:1–58:8. 1

[KGJVM83] Kirkpatrick S., Gelatt Jr C., Vecchi M., McCoy A.: Optimization by Simulated Annealing. *Science 220*, 4598 (1983), 671–679. 6

[KS86] Kender J. R., Smith E.: Shape from darkness: Deriving surface information from dynamic shadows. In *AAAI* (1986), pp. 664–669. 2

[KvWW09] Keiren J., van Walderveen F., Wolff A.: Constructability of trip-lets. In *Abstracts from the 25th European Workshop on Computational Geometry, Brussels, Belgium* (2009), pp. 251–254. 2

[MP09] Mitra N. J., Pauly M.: Shadow art. *ACM Transactions on Graphics 28*, 5 (Dec. 2009), 156:1–156:7. 2

[NBB04] Nayar S. K., Belhumeur P. N., Boult T. E.: Lighting sensitive display. *ACM Transactions on Graphics 23*, 4 (Oct. 2004), 963–979. 1

[PQW*08] Pang W.-M., Qu Y., Wong T.-T., Cohen-Or D., Heng P.-A.: Structure-aware halftoning. *ACM Transactions on Graphics 27*, 3 (Aug. 2008), 89:1–89:8. 3, 6

[RPL89] Raviv D., Pao Y.-H., Loparo K.: Reconstruction of three-dimensional surfaces from two-dimensional binary images. *IEEE Transactions on Robotics and Automation 5*, 5 (1989), 701–710. 2

[SK83] Shafer S. A., Kanade T.: Using shadows in finding surface orientations. *Computer Vision, Graphics, and Image Processing 22*, 1 (1983), 145 – 176. 2

[YC02] Yu Y., Chang J. T.: Shadow graphs and surface reconstruction. In *Proceedings of the 7th European Conference on Computer Vision-Part II* (2002), pp. 31–45. 2