

Hiding a Second Appearance in a Physical Relief Surface

Yi-Liu Chao and Daniel G. Aliaga

Department of Computer Science, Purdue University, West Lafayette, Indiana, USA
{chao1, aliaga}@cs.purdue.edu

Abstract. We present a novel information hiding process that couples geometrical modeling with automated 3D fabrication for creating hidden-appearance reliefs. Our relief surface produces a first grayscale appearance visible by simple direct illumination and a second grayscale appearance ensured to be visible when the relief is lit by a digital projector with a specifically designed pattern and from a particular direction. The two appearances/images can be different yet embedded in the same physical relief. Since the second appearance appears only on demand, it could be used to hide a second image, a company logo, or a watermark image, for instance. Our novel method calculates a relief surface that maintains the properties needed for producing a second (hidden) appearance while also ensuring the first appearance is visible under normal direct illumination. Our experiments show that our method robustly produces reliefs with two arbitrary desired appearances.

Keywords: information hiding, images, reliefs, surfaces, watermarks, 3D manufacturing .

1 Introduction

In this paper, we present a novel application of information hiding whereby two visual appearances (or “images”) are encoded into a single physical relief surface. Our work exploits advances in digital manufacturing but focuses on a computational modeling component. We wish to design a physical *relief* surface to have a first appearance visible to the naked eye under normal directional illumination and defined by a provided arbitrary image (Figure 1, shaded image A). In addition, we wish the same relief to have a second appearance, defined by an arbitrary second image, which is made visible only when the relief is lit by a carefully designed illumination pattern from a specific location (e.g., by using a digital projector) (Figure 2, shaded image B). Since the second appearance can be made to appear only on demand, it could be used to hide a second image, a company logo, or a watermark image, for instance. The second image is secured since it is only retrievable by knowing both the position of the projector and the carefully-designed illumination pattern. To our knowledge, there is no previous information hiding approach as ours. Some previous works do incorporate multiple appearances into a relief/object, however our novel process ensures the second appearance is always possible despite potential self-shadows and the implicit finite projector light radiance. In the absence of our method, the two appearances/images are not always possible (Figure 1, bottom row). We anticipate our methodology will lead to significant more work in this exciting novel application.

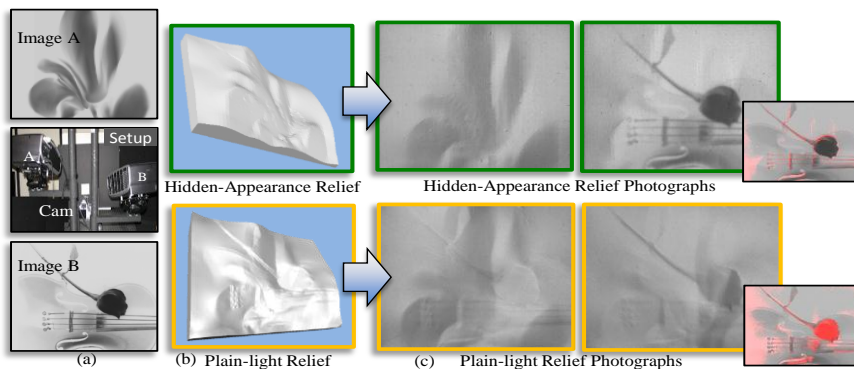


Fig 1: Hidden-Appearance Reliefs. a) We create a relief that produces image A under normal direct illumination and image B only under projector illumination. b) We show our hidden relief geometry (top) and a plain-light relief geometry (bottom) – an implementation of [1]. c) Photographs of two physical relief surfaces. Left pair is under normal direct illumination and yielding image A; right top is image B with our approach and right bottom is image B with plain-light relief geometry. The small insets show per pixel intensity error between the photographs and the desired image B (red = large error). Our method is able to produce both images A and B, despite them being very different.

Previously, papers have addressed generating surfaces with purposefully encoded data and/or purposefully crafted visual behaviors. In the synthetic world, a relevant set of works are algorithms which robustly or fragily encode watermarks into the digital representation of the mesh (e.g., [2]). In the world of digitally manufactured physical objects, methods have encoded fragile marks (e.g., for genuinity detection) into physical surfaces [3] or into paper [4], and have manufactured surfaces yielding a pre-specified shading, or appearance, behavior (e.g., [5, 6]).

Our work is inspired by an observation in Chen et al. [7] which states that given any two appearances for a single diffuse surface there is, in theory, always a combination of surface geometry, albedo patterns, and light sources that can produce the appearance pair. In practice, the limited amount of light, manufacturing restrictions on heightfield sharpness, and self-shadows imposes practical restrictions on the images. However, we have found that a wide range of imagery is possible with our method. More concisely, our methodology for generating a hidden appearance in a relief is based on the following three key observations:

- *there are multiple relief geometries that yield the same shaded image when viewed from above the surface*; we exploit the multiplicity of solutions to find a combination of surface heights that produces both shaded image A and shaded image B;
- *if the relief were not designed to explicitly support/hide the second appearance, then the second image cannot in general be produced*; this is true even with the help of a projector emitting any desired illumination pattern; and
- *the use of an illumination pattern for generating the second appearance enables using a constant albedo to produce any two grayscale shaded images A and B*; image A is visible to the naked eye; however, since the albedo is constant (e.g., no paint or material change is visible to the naked eye), image B is only visible to the naked eye by using the proper illumination pattern; the

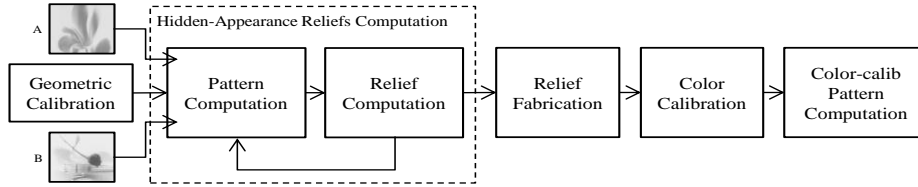


Fig 2: System Pipeline. A summary of the major components of our method.

pattern itself could, for example, be encoded, or generated, by a key-based procedure and our overall methodology ensures an arbitrary chosen second appearance is possible.

Our approach uses an optimization process to create a diffuse surface heightfield that will be subsequently manufactured. Creating a relief surface that produces the appearance of a single shaded image A under normal direct illumination is simple: a set of normals are computed via optimization and then integrated to give a heightfield. We also wish to ensure an arbitrary shaded image B is visible when illuminated by a digital projector shining a particular illumination pattern. A suitable combination of surface geometry and illumination pattern are initially unknown and must be determined. Further, since the projector illumination range is limited, the computed pattern must also be constrained to lie within possible illumination range. A straightforward formulation results in a large set of constrained inequality equations that is nonlinear in the unknowns – solving this system is both impractical and highly non-robust. Instead, we perform several simplifications that result in an efficient solution using equations that are linear with respect to the unknowns and include linear smoothing equations and linear constraints. A suitable relief can be computed, tested in simulation, and then fabricated.

We have implemented a complete prototype system to produce *hidden reliefs* (Figure 2). The relief is automatically manufactured from our computed model using a 3D printer. The relief is placed on a stage in front of a digital camera and two digital projectors: one projector is used as a simple point light source to directly illuminate the relief so as to produce image A, and the other projector emits the illumination pattern for image B. The camera and projectors have been geometrically and radiometrically calibrated beforehand. Our results include the design and fabrication of several two appearance reliefs, and theoretical and empirical comparisons to previous related works (e.g., Alexa and Matusik [1] -- we call them *plain-light reliefs*) and to reliefs created for a single appearance but using projector patterns to obtain the second appearance (we call these *single-appearance reliefs*). Our experiments consistently demonstrate that using our approach yields an improved ability to encode both image A and image B into a single relief.

2 Related Work

Information hiding can be viewed as an exploitation of flexibility and, in some cases, redundancy. With this in mind, the concept of watermarking has been extended to the digital domain. Abundant literature investigates watermarks in digital images [8, 9] and in digital audio files [10]. It has been used to seamlessly hide watermarks in

3D meshes (e.g., [2, 11, 12]). Approaches are concerned with robustness, security, or both (e.g., [13]). These methods have been designed for digital data which can be created, read, and replicated with zero error while physical objects cannot be replicated with zero error.

People have worked on hiding information on physical surfaces. Aliaga et al. [3] brought the watermark concept to physical surfaces. They fabricate relief surfaces encoding watermarks such that physically copying the watermark is hard. While, similar to these works, we wish to embed information, we seek to do so in the visual appearance of a physical relief surface. Some previous surface and relief work has attempted to encode multiple appearances. For example, Oliva et al. [14] design a colored pattern (i.e., single flat colored image) which gives the illusion of a different appearance at different viewing distances. Alexa and Matusik [1] use constant albedo and alter the surface geometry so as to produce a different image when directly illuminated from one of two different directions. However, in both works the two involved images cannot be arbitrary -- they must be designed to work well together.

Although, we could hide a second appearance by secretly using relief materials or unbeknownst reflective paint patterns, we seek an “open method”, closer to Kerckhoff’s principle [15]. Our methodology assumes a public method and a single albedo (i.e., a single material and color) yet is still enable to hide an arbitrary second appearance (e.g., one that can be procedurally generated based on a key), and is visible only when appropriately illuminated. Additional surface hiding methodologies can be viewed as complementary.

In contrast, our method yields two novel abilities. First, shaded image A and shaded image B can be arbitrary, very different, gray-shaded images. In fact, since we are using a projector to produce image B, it can even include the physically impossible shaded images described by Horn et al. [16], without affecting image A. Second, shaded image B is only visible when appropriately illuminated. While a geometrically and chromatically calibrated projector illumination system can impart a new appearance on physical surfaces (e.g., state-of-the-art report [17]), it is not sufficient to yield an arbitrary image A and image B. In particular, even with perfect calibration an arbitrary image B cannot be produced. Rather, the surface geometry must be altered so as to ensure image B can be produced by an illumination pattern while subject to the constraint of ensuring image A is what is visible under normal direct illumination of the relief – such a surface geometry is precisely what our method computes. Although, we do not explicitly maximize the imperceptibility of image B under normal illumination, its existence is not evident to the naked eye; in our results section, we do analyze the impact of the contrast and sharpness of image A and B on the hiding of image B. Collectively, these abilities lead to novel applications; for example, embedding a watermark into a physical relief or a desired alternative appearance suitable for other image processing (e.g., object tracking).

3 Hidden-Appearance Reliefs

The construction process for our reliefs iteratively finds a single surface that supports the two desired appearances. First, we describe the physical setup and present an

appearance formation process for hidden-appearance reliefs. Then, we simplify the formulation and describe an iterative optimization process and smoothing equations.

3.1 Relief Setup

Our setup consists of the relief surface observed from above (i.e., viewing direction of $[0,0,-1]$) and two projectors (Figure 3a). A first projector is used to generate the directional light for producing image

A. It is positioned along a direction l_A at a small angle to the viewing direction. A second projector is along the direction l_B at a larger angle to the viewing direction and is used to shine the illumination patterns for generating image B.

Since the projector, camera, and relief mesh are usually of different resolutions, we choose to define computations in terms of each triangle $i \in [1, N]$ of the relief mesh M and project the triangle to the calibrated camera and projector image planes in order to calculate desired image intensity values and projector pattern values. This also enables us to control computational cost by altering the resolution of mesh M . In order to produce a symmetric mesh (i.e., one that is equivalent upon a rotation of 90, 180 or 270 degrees), we add an additional vertex in the middle of the quadrilaterals of a standard rectilinear meshing of vertices (Figure 3b). Furthermore, we assume M to be a heightfield over the XY plane, thus only the z -coordinates of the mesh vertices are free to move.

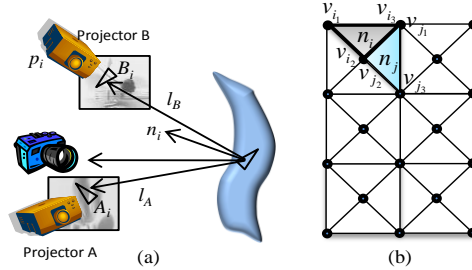


Fig 3: Setup and Relief Mesh. a) A diagrammatic view of the relief, projectors, and camera. b) The symmetric triangulation of the relief mesh. Both diagrams are labeled with the variables used in our formulation.

3.2 Appearance Formation

Our formulation of the appearance formation process uses a diffuse reflectance model to express the behavior of the relief mesh M under the desired two illumination scenarios. Figure 4 contains an intuitive and synthetic 2D example. Figure 4a shows a challenging pair of 1D image A and B - image B is chosen as the “opposite image” of A, coincidentally an impossible shaded image as per Horn et al. [16]. The direct illumination for image A is from directly above and the pattern for producing image B is illuminated slightly from the right side. Figure 4b

shows four surfaces (i-iv) that all yield image A; e.g., in the leftmost surface (i) the

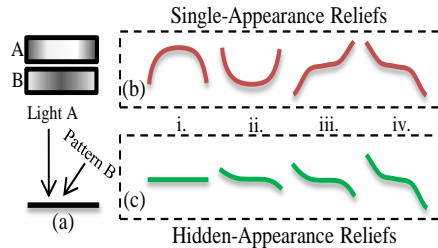


Fig 4: Image Formation Example. a) Desired images A and B. b) Four possible reliefs (i-iv) that produce image A but not necessarily image B even when using a projector. c) Our optimization iteratively finds a relief surface able to produce both A and B (steps i-iv).

amount of reflected light is maximal in the middle and falls off to the sides, as in image A. The non-uniqueness of the solution for A is precisely what we exploit. However, these four meshes are not necessarily able to produce image B. Again, consider the leftmost surface in 4b: to yield a bright left-side of the surface, the projector must shine a large amount of light that might exceed the maximum illumination ability of the projector (even when disregarding self-occlusions). Figure 4c shows our approach which begins with a flat surface and iterative finds the relief surface (i-iv) that produces image A and also is able to produce image B with the help of a projector. In a full-fledged example, our iterative process concurrently finds over the entire surface a heightfield configuration able to produce both image A and B.

For producing appearance A, the image formation process for triangle i can be expressed by the well-known equation

$$\alpha(n_i \cdot l_A) = A_i \quad (1)$$

where α is the constant surface albedo of M , n_i is the desired normal of relief triangle i , and A_i is the mean intensity of the pixels in A onto which triangle i projects.

To express the second appearance, we use the inequality

$$\alpha(n_i \cdot l_B) \geq B_i \quad (2)$$

where B_i is the mean intensity of the pixels in B onto which triangle i projects. Equation (2) ensures that for surface normal n_i at least the intensity needed to yield B_i is possible. An inequality is appropriate because ultimately an illumination pattern is used and the pattern values can reduce the amount of incident light (but not increase it).

By explicitly factoring in the illumination patterns, the inequality in (2) can be converted to the following equality

$$\alpha p_i(n_i \cdot l_B) = B_i \quad (3)$$

where p_i is the mean intensity value of the projector pixels that cover relief triangle i .

Our approach seeks relief triangle normals n_i and bounded projector pattern values $p_i \in [0,1]$ that simultaneously satisfy equations (1) and (3) for all triangles $i \in [1, N]$. We denote vertices of triangle i as $v_{i_1}, v_{i_2}, v_{i_3}$ and write n_i in terms of the normalized cross product of the vertices:

$$\alpha \left(\frac{(v_{i_2} - v_{i_1}) \times (v_{i_3} - v_{i_1})}{\|(v_{i_2} - v_{i_1}) \times (v_{i_3} - v_{i_1})\|} \cdot l_A \right) = A_i \quad (4)$$

$$\alpha p_i \left(\frac{(v_{i_2} - v_{i_1}) \times (v_{i_3} - v_{i_1})}{\|(v_{i_2} - v_{i_1}) \times (v_{i_3} - v_{i_1})\|} \cdot l_B \right) = B_i$$

which is a nonlinear expression with respect to the unknowns (i.e., the z coordinates of the triangle vertices and the pattern values p_i). The complete equation set defined by (4) constitutes a large (though sparse) nonlinear irrational equation system that is difficult to solve.

3.3 Simplification

In order to efficiently and robustly solve for the relief mesh height values and projector pattern values, we use simplifying heuristics. Using the heuristics results in two sets of linear equations solved in an alternating fashion in order to incrementally find pattern values and relief heights, and to collectively approximate equations (4).

Our first heuristic is to assume the length of the triangle normals n_i are constant during an iteration and thus remove the square root in the denominator of equation (4). This rewrites equation (4) as

$$\frac{\alpha}{\|\hat{n}_i\|} \begin{bmatrix} y_{i_{21}}z_{i_{32}} - y_{i_{32}}z_{i_{21}} \\ z_{i_{21}}x_{i_{32}} - z_{i_{32}}x_{i_{21}} \\ x_{i_{21}}y_{i_{32}} - x_{i_{32}}y_{i_{21}} \end{bmatrix} \cdot l_A = A_i \quad (5)$$

$$\frac{\alpha}{\|\hat{n}_i\|} p_i \begin{bmatrix} y_{i_{21}}z_{i_{32}} - y_{i_{32}}z_{i_{21}} \\ z_{i_{21}}x_{i_{32}} - z_{i_{32}}x_{i_{21}} \\ x_{i_{21}}y_{i_{32}} - x_{i_{32}}y_{i_{21}} \end{bmatrix} \cdot l_B = B_i \quad (6)$$

where $\|\hat{n}_i\| = \|(v_{i_2} - v_{i_1}) \times (v_{i_3} - v_{i_1})\|$. We write $v_{i_2} - v_{i_1}$ as $[x_{i_{21}} \ y_{i_{21}} \ z_{i_{21}}]^T$ and $v_{i_3} - v_{i_1}$ similarly. Since $\|\hat{n}_i\|$ is considered a constant during an iteration and the x and y coordinates are fixed, equation (5) is a linear equation of the unknowns z_{ij} 's. Equation (6) is not yet linear because of the multiplication by the unknown p_i . We use the following heuristic to further simplify the problem.

The second heuristic is to assume a current estimate either for mesh geometry M or for the pattern values p_i . This produces two formulations of equations (5) and (6).

- i. The first formulation solves for p_i 's using linear equations by assuming a known geometric mesh (i.e., all z 's are constant). Equation (6) is used because no p_i term appears in equation (5). p_i values are restricted to the range $[0,1]$ by using a constrained linear optimization.
- ii. The second formulation solves for the mesh heights by assuming constant values for p_i . This formulation uses both equations (5) and (6) which are now both linear in the unknowns (i.e., the z values of the vertices) and can be solved using linear optimization.

The full equations set are in general over-constrained but relatively sparse. Since a vertex is used in up to only eight adjacent triangles, the system of equations is always fairly sparse. Hence, a sparse (constrained) linear least squares optimization can solve formulation (i) or (ii) relatively quickly, even for a large number of mesh triangles.

3.4 Iterative Optimization

To compute the relief mesh, we iterate between solving for pattern values and for geometry mesh heights until converging to a final surface. The validity of our equations only holds for small height value changes. In particular, as the vertex heights change, the constant length of n_i 's, the values of the p_i 's, the triangle to projector correspondence used to calculate p_i 's position on the projector image plane, and the triangle to camera correspondence used to compute the A_i 's and B_i 's pixel intensity become increasingly inaccurate. Hence, our optimization starts with a planar

relief mesh and computes an initial set of pattern values p_i using formulation (i). Then, pattern values p_i are used to obtain new vertex heights using formulation (ii). The new mesh updates correspondence of relief, camera, and projectors, and triggers re-computing the pattern values and triangle normal lengths for the next iteration.

From a theoretical standpoint, our iterative optimization process and equation sets do not guarantee that a solution will be found nor that it is unique. Rather, we find a surface that satisfies the specified shading behavior in a least squares sense. In practice however, we found approximate solutions to always exist.

3.5 Smoothness

To provide support for ensuring incremental changes to the surface, for increased robustness, and for the creation of an approximately smooth surface (beneficial to physical manufacturing), we include additional equations. These equations ensure that the height changes of neighboring vertices are similar in one iteration. We define such an equation set for all edges in mesh M .

We rewrite the equations in terms of height changes in one computation in order to ensure similar variations of neighbors during an iteration. We denote with Δz_j the height change of the j th vertex of triangle i in one computation and we use Δz_u and Δz_v to represent height changes of two mesh vertices where the edge (u, v) is in relief mesh M . We rewrite equations (5) and (6) in terms of height changes and we incorporate a smoothness requirement. Altogether the per-iteration task is to minimize the following expression:

$$\begin{aligned} & \sum_{i \in [1, N]} \left[E_{Ai} \cdot \begin{bmatrix} \Delta z_{i_1} \\ \Delta z_{i_2} \\ \Delta z_{i_3} \end{bmatrix} - \left(\frac{\|\hat{n}_i^{(k)}\|}{\alpha} A_i - \hat{n}_i^{(k)} \cdot l_A \right) \right]^2 + \\ & \sum_{i \in [1, N]} \left[E_{Bi} \cdot \begin{bmatrix} \Delta z_{i_1} \\ \Delta z_{i_2} \\ \Delta z_{i_3} \end{bmatrix} - \left(\frac{1}{p_i} \frac{\|\hat{n}_i^{(k)}\|}{\alpha} B_i - \hat{n}_i^{(k)} \cdot l_B \right) \right]^2 + \quad (7) \\ & 1 / \beta \sum_{(u,v) \in M} [\Delta z_u - \Delta z_v]^2 \end{aligned}$$

where

$$E_{Ai} = \begin{bmatrix} l_{Ax}(y_{i_3} - y_{i_2}) + l_{Ay}(x_{i_2} - x_{i_3}) \\ l_{Ax}(y_{i_1} - y_{i_3}) + l_{Ay}(x_{i_3} - x_{i_1}) \\ l_{Ax}(y_{i_2} - y_{i_1}) + l_{Ay}(x_{i_1} - x_{i_2}) \end{bmatrix}$$

and $l_A = [l_{Ax} \ l_{Ay} \ l_{Az}]^T$ represents the normalized direction vector of light source A. E_{Bi} and l_B are defined similarly for light source B (i.e., that from the projector). The term $\hat{n}_i^{(k)} = (v_{i_2}^{(k)} - v_{i_1}^{(k)}) \times (v_{i_3}^{(k)} - v_{i_2}^{(k)})$ is the normal vector computed from the height values of the mesh during iteration k -- they are constant during an iteration.

The first part of equation (7) defines the relief appearance formation objective for all mesh triangles and the second part the smoothing desire. The $1/\beta$ term is a normalizing factor used to balance the relative importance of the image formation equations and the smoothing equations. In practice, we compute β so as to provide equal importance to image formation and to smoothing. Since the number of equations used in image formation is about half of those used for smoothing, we usually set $\beta = 2$.

4 Implementation Details

Our prototype system includes geometric and color calibration. Geometric calibration of the 1400x1050 resolution Optoma DLP projectors and the 10MP Canon Rebel XTi camera is done once. Color calibration is recomputed for each fabricated object [18]. The fabrication process is automated using our Alaris30 3D printer. After fabrication, we place the object in front of our camera and projectors on a platform that can be mechanically repositioned using knobs (middle of Figure 1(a)). To place the object accurately at the origin of the calibrated camera-projector coordinate frame, the projector illuminates a contour light pattern which is then used to manually align the object with the contour lit by the projectors.

5 Analysis of Intensity Coverage

We have analyzed the theoretical intensity ranges achievable for any given image pair (A,B). Our method can obtain a large range of intensity differences between A and B images, in fact more than the plain-reliefs of Alexa and Matusik's [1] (Figure 5). In particular, our method supports all lower intensity values for image B.

For the analysis, we focus on measuring the intensity of a plane since our mesh consists of triangles. We assume the simple light source direction to be l_A and the specifically designed digital projector light direction to be l_B . We are looking for a triangle i which has a normal n_i that satisfies the following two equations:

$$n_i \cdot l_A = A_i \quad \text{and} \quad p_i n_i \cdot l_B = B_i \quad (8)$$

where $p_i \in [0, 1]$ is the intensity of the incident projector from l_B , A_i is the desired intensity of the triangle when lit by the directional light from l_A , and B_i is the desired intensity of the triangle when lit by the designed light pattern from l_B . Hence, A_i and B_i is achievable if

$$n_i \cdot l_A = A_i \quad \text{and} \quad n_i \cdot l_B \geq B_i. \quad (9)$$

Geometrically, these equations define two cones shown in Figure 5(a). A_i is achievable when n_i falls exactly on the surface of a cone defined by the first equation. B_i is achievable when n_i falls inside the cone defined by the second inequality. The second equation is always achievable when $B_i = 0$. As shown in Figure 5(a), the light directions l_A and l_B define the centerlines of the cones. Let θ be the angle between l_A and l_B . Intensity values A_i and B_i define the angles γ

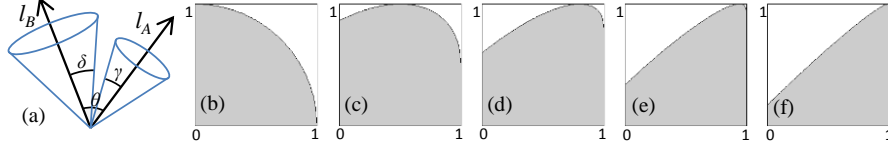


Fig 5: Intensity Coverage. a) Setup used in b-f. b-f) Supported intensity coverage for images A & B varies under different light directions. Light directions in b-f are defined as $l_A = (1 + h^2)^{-0.5} [1 \ 0 \ h]$ and $l_B = (1 + h^2)^{-0.5} [0 \ 1 \ h]$ with $h = \{0, 1, 2, 4, 8\}$. x-axis represents image A intensity and y-axis represents image B intensity. A point (x, y) is gray when it is possible to find a normal that shines x under simple light from l_A and shines y under designed illumination from light direction l_B .

and δ , the angles between the cone surfaces to cone centerlines l_A and l_B . n_i has a solution as long as any part of the cone centered around l_A falls inside the other cone; i.e., when $\delta \geq \theta - \gamma$. Hence, given light directions l_A and l_B and a particular value for γ (or A_i), there is a range of δ (or B_i) which produces at least one solution for n_i .

We show in Figures 5b-f the pairs of values for A_i and B_i that are possible for several light directions. In each of figures 5b-f, the x-axis is a value for $A_i \in [0, 1]$ and y-axis is a value for $B_i \in [0, 1]$. The angle θ between light directions varies from 90 to 0 degrees from (b) to (f). For a specific A_i value a , we draw a vertical gray segment along line $x = a$ to show what range of B_i values are achievable as per equation 9; in other words, a point (a, b) falling in the gray area implies that the intensity pair $A_i = a$ and $B_i = b$ is achievable. As observed, our method supports a larger set of intensity than previous work (i.e., [1]) since we cover all lower intensities in image B due to the simultaneous optimization of surface shape and projector light.

6 Results and Discussion

We have used our approach to design several relief surfaces both in simulation and in real-life. We used tessellated meshes of resolution 100x80 cells which require a compute time of about 30 minutes (about half of that time is spent in actual optimization computations and the rest in file I/O). Our typical 3D print time is 5-10 hours for 10x8 centimeter reliefs.

In Figure 1, we show photographs of an example hidden relief mesh and a plain light relief mesh produced by [1]. The latter relief mesh uses only simple lighting and is designed to yield image A when illuminated from one direction and image B when illuminated from a different direction. We use the same light directions for both reliefs. As seen, the relief mesh of [1] is not able to produce both appearances – this is mostly because of the significant intensity differences between images A and B (see Section 5). In contrast, our approach can produce both appearances quite well. The visualization inset on the right shows a color-coded image of the errors of both reliefs. Note that even though we take image B into account when computing the hidden-appearance relief, image B is not perceivable in the relief under normal illumination.

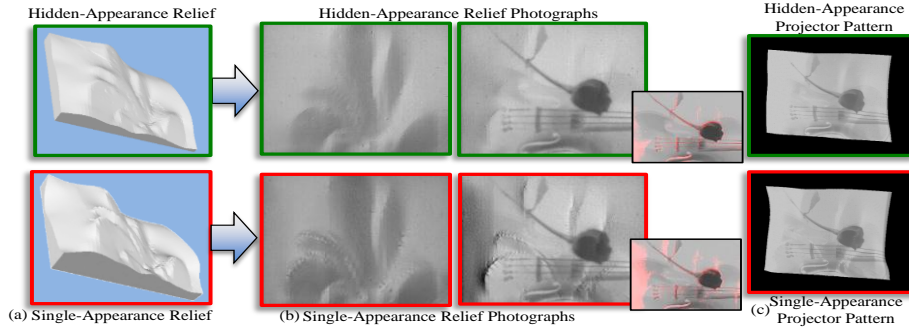


Fig 6: Hidden- vs Single-Appearance Reliefs. a) We show our hidden-appearance relief geometry (top) and a single-appearance relief geometry (bottom) which produces appearances of image A and B as shown in Figure 1. b) Photographs of our hidden-appearance relief under direct illumination yielding image A (top middle) and under projector light yielding image B (top right). A single-appearance relief is able to produce image A (bottom middle) but the second appearance cannot necessarily be produced even with the help of a digital projector (bottom right). The small insets are a visualization of the per pixel intensity error between the photographs and image B (red = large error). c) The projector patterns that shine on the two reliefs when producing image B.

In Figure 6, we compare our hidden-appearance relief to a single-appearance relief using the same image content as in Figure 1. The geometry of a single-appearance relief is computed for only one appearance (image A). Then, we compute the projector pattern that best achieves the second appearance (image B). Our approach is able to faithfully recreate both appearances despite both relief types using projectors. As seen in Figures 6c, the projector patterns for both relief types are similar. This means the need for the projector pattern is roughly equal in both cases. While one naïve option to produce appearance B would be to shift all the content to the projector pattern, it would require simplifying the relief geometry to nearly a plane. This would violate the desire to have appearance A be produced by a simple directional light. Instead, our optimization process finds a geometry able to produce image B, with the aid of a projector, while leaving the appearance of image A intact.

Figure 7 contains several instances of a hidden-appearance relief each created in simulation with a different mesh resolution. The lowest resolution mesh (50x40 cells,

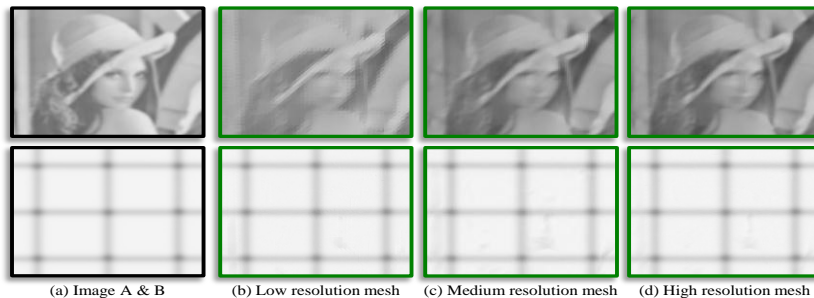


Fig 7: Hidden-Appearance Reliefs with Different Mesh Resolutions. We experiment with altering mesh resolution. a) Shows image A (top) and image B (bottom). Appearances resulting from using b) low resolution mesh with 50x40 cells, c) medium resolution mesh with 100x80 cells, and d) high resolution mesh with 150x120 cells.

10 minutes total compute time) shows noticeable visual artifacts and blurriness as compared to the highest resolution mesh (150x120 cells, 50 minutes). We found the mesh resolution of 100x80 cells (and 30 minutes total compute time) to be a reasonable balance of visual quality and computation time.

In order to better understand what type of images we can hide, we experimented in simulation with the effect of varying contrast levels and sharpness in image A and/or B. When A has a small contrast, the resulting relief only needs low frequency height changes and thus tends to be flat. It is easy for the projector to shine the patterns needed to produce image B. In short, low contrast in A makes the hidden-appearance relief problem easy. A similar effect occurs with a low contrast B image as well. When B has low contrast, even though the relief is not optimized for the image, the projector can do a lot to compensate for the undulations of the relief surface. In Figure 8 we show that when both A and B have low contrast, both hidden-appearance relief and single-appearance relief surfaces do a good job of producing an image B -- the problem itself is fairly easy. However, when A has high contrast, the relief surface needs significant height changes to produce image A under normal illumination. Hence, it is easy to unwillingly obtain a surface for which it is hard for the projector to alter the appearance to produce image B -- even self-shadows occur more readily. Moreover, if B has high contrast, it makes the inequality equation (2) even harder to achieve. As long as the inequality is not satisfied then B is not achievable given limited projector power. In Figure 8, we show the results of single-appearance reliefs and hidden-appearance reliefs with high contrast A and B images. As seen, the single-appearance relief must do a significant effort to achieve A, which generates a bumpy surface geometry and easily breaks the generation of image B. However, our hidden-appearance reliefs produce image B while maintaining the appearance of image A under normal illumination (and also not showing image B). Hence, although any second appearance can be encoded, using images A and B of relatively high contrast increases the benefit of using our method to hide the second appearance – a naively generated surface for the same images

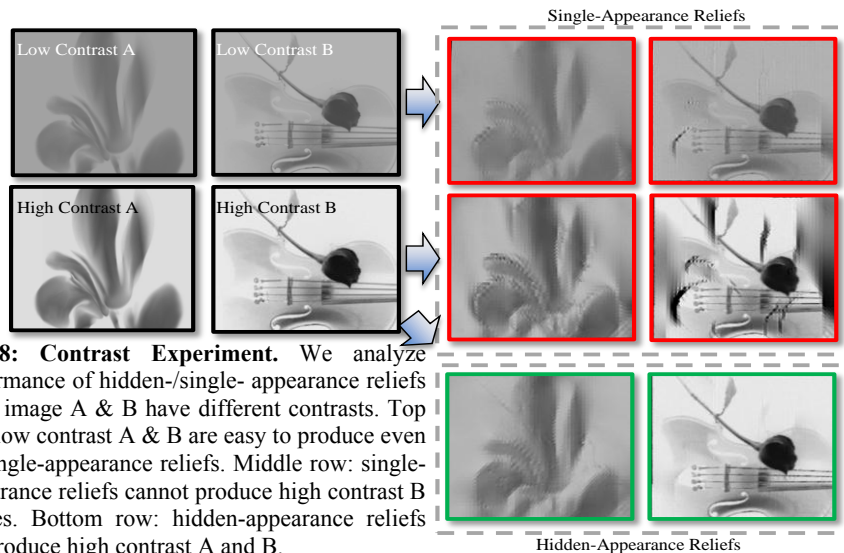


Fig 8: Contrast Experiment. We analyze performance of hidden-/single- appearance reliefs when image A & B have different contrasts. Top row: low contrast A & B are easy to produce even for single-appearance reliefs. Middle row: single-appearance reliefs cannot produce high contrast B images. Bottom row: hidden-appearance reliefs can produce high contrast A and B.

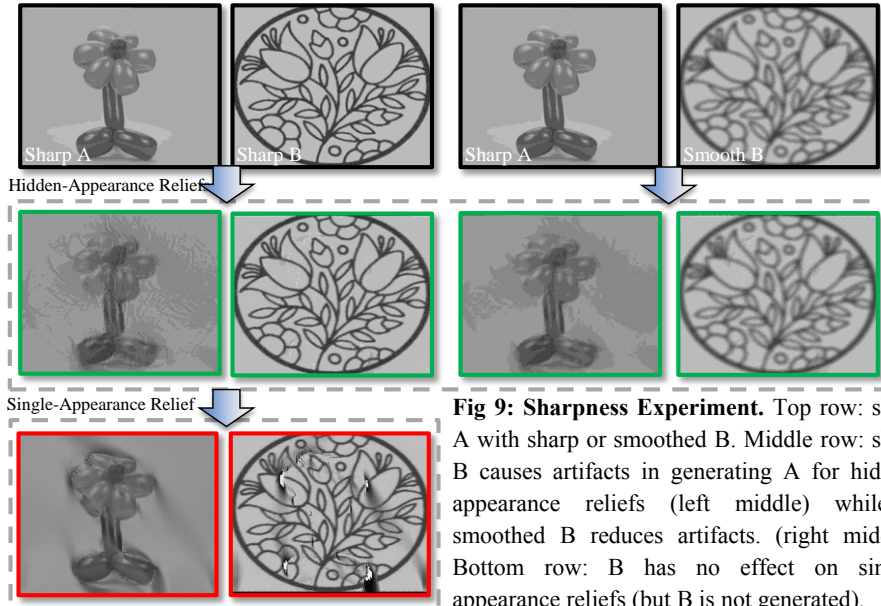
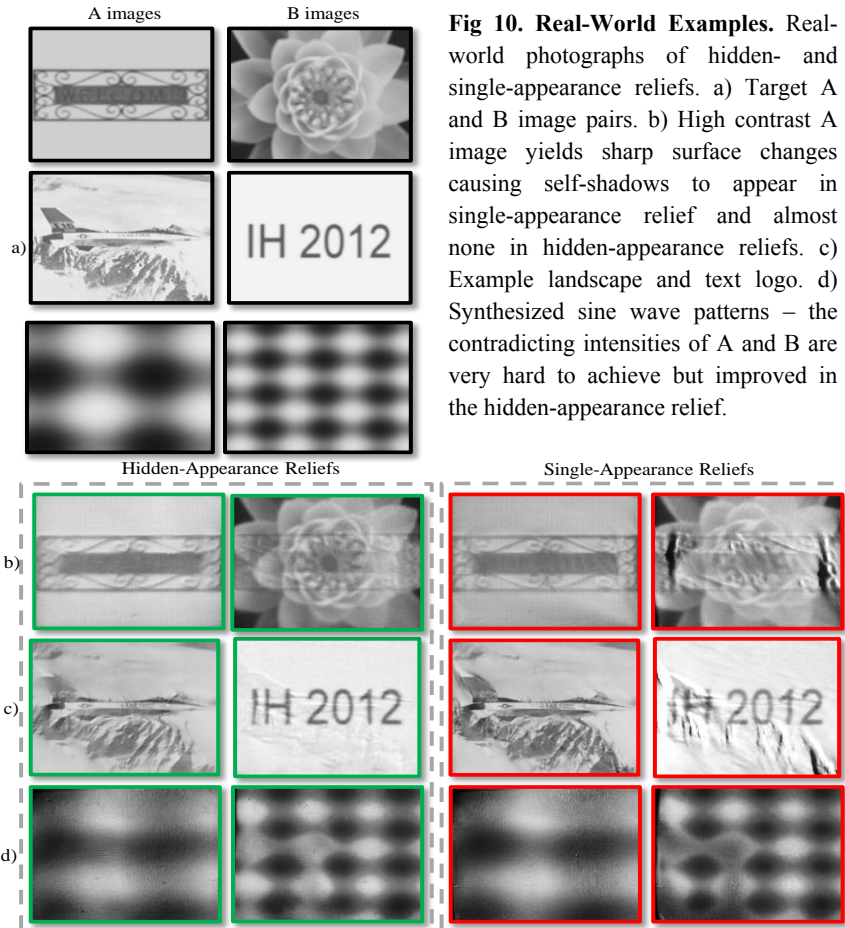


Fig 9: Sharpness Experiment. Top row: sharp A with sharp or smoothed B. Middle row: sharp B causes artifacts in generating A for hidden-appearance reliefs (left middle) while a smoothed B reduces artifacts. (right middle). Bottom row: B has no effect on single-appearance reliefs (but B is not generated).

would not reproduce image B as well.

In Figure 9, we analyzed in simulation the effect of sharpness in the input image A and/or B. When A is sharp, the height values need to change a lot to achieve the sharp image. This makes it difficult for both single- and hidden-appearance reliefs. Nevertheless, sharpness in B does not have any effect on single-appearance reliefs because it does not consider B at all. Generation of A in single-appearance relief is not affected by sharpness in B (bottom row of Figure 9). However, sharpness in B causes noise in hidden-appearance relief geometries. This effect results from the simplification we made about correspondences: we assume that the correspondences between mesh vertices and camera pixels do not change when the geometry change is small. However, when B is sharp, small changes in correspondences may cause large changes in corresponded intensity. It could be that one relief triangle is asked to have a white appearance in one iteration and a black appearance in the next iteration. This causes the noisy artifacts in left of middle row in Figure 9. Hence, to obtain a geometry that does not show remnants of image B under normal illumination, better results are achieved with a smoothed B, (right of middle row in Figure 9).

Figure 10 shows several real world experiments and photographs. We show hidden-appearance reliefs and single-appearance reliefs for various A and B image pairs. For each, we compute the hidden- and single-appearance relief, fabricate them, color calibrate them, compute the color calibrated pattern, and capture photographs of the physical object. Figures 10b-d show photographs of hidden-appearance reliefs producing images A and B better than single-appearance reliefs. In particular, Figure 10c shows the challenging case of sharp images A and B (see discussion for Figure 9) and Figure 10d shows a particularly hard example where A and B systematically contradict each other in their visual objectives. Nevertheless, our method shows notable improvement.



7 Conclusion and Future Work

We present hidden-appearance reliefs which enable a chosen appearance A to be observed under direct lighting while also enabling a hidden arbitrary appearance B which is only visible under a particular lighting setup. By doing this, we effectively hide a second piece of information into one single relief. We present a computational method which designs hidden-appearance reliefs and a full implementation. Our experiments show that our method is robust for various A and B image pairs both in simulation and in real-life.

Our future works includes the following. 1) We would like to further analyze the security aspect of our method. For instance, how secure is the second image? Is it possible to retrieve the second image with reverse-engineering? 2) We would also like to extend our method to include colored images. 3) Although the second appearance is not recognizable under normal illumination in our current approach, we do not

explicitly guarantee, or maximize, that it is unrecognizable when observed under normal illumination or when illuminated by a pattern other than the indicated one. Thus, we seek an extension that models the perceptibility of the second appearance and purposefully attempts to keep it small. 4) Another interesting extension is to incorporate multiple B appearances produced by different projector pattern illuminations and to quantify the “amount of information” that can be hidden.

References

1. Alexa M., Matusik W.: Reliefs as Images. In: ACM Trans. on Graphics, 29(4) (2010)
2. Wang K., Lavoue G., Denis F., Baskurt A.: 3D Meshes Watermarking: Review and Attack-Centric Investigation. In: International Workshop on Information Hiding, 50-64 (2007)
3. Aliaga D., Atallah M.: Genuinity Signatures: Designing Signatures for Verifying 3D Object Genuinity. In: Computer Graphics Forum (Eurographics), 28(2), 437-446 (2009)
4. Volpe H.R.: Printing method and copy-evident secure document. US Patent 5487567, January (1996)
5. Weyrich T., Peers P., Matusik W., and Rusinkiewicz S.: Fabricating Microgeometry for Custom Surface Reflectance. In: ACM Transactions on Graphics, 28(3) (2009)
6. Weyrich T., Deng J., Barnes C., Rusinkiewicz S., Finkelstein A.: Digital Bas-Relief from 3D Scenes. In: ACM Trans. on Graphics, 26(3) (2007)
7. Chen H., Belhumeur P., Jacobs D.: In search of illumination invariants. In: IEEE CVPR, 254–261 (2000)
8. Ryu S., Lee M. and Lee H.: Detection of Copy-Rotate-Move Forgery Using Zernike Moments. In: Information Hiding Conference, 51-65 (2010)
9. Schwamberger V., Le P.H.D., Scholkopf B., and Franz M.O.: The Influence of the Image Basis on Modeling and Steganalysis Performance. In: Information Hiding Conf. (2010)
10. Arnold M., Chen X., Doerr G., Baum P.: Improving Tonality Measures for Audio Watermarking. In: Information Hiding Conference (2010)
11. Kim K., Barni M., Tan H.: Roughness-Adaptive 3D Watermarking of Polygonal Meshes. In: Information Hiding Conference (2009)
12. Wang K, Lavou'e G., Denis F., and Baskurt A.: Three-Dimensional Meshes Watermarking: Review and Attack-Centric Investigation. In: Information Hiding Conference (2007)
13. Cao Y., Zhao X., Feng D., Sheng R.: Video Steganography with Perturbed Motion Estimation. In: Information Hiding Conference (2011)
14. Oliva A., Torralba A., Schyns P.: Hybrid Images. In: ACM Trans. on Graphics, 25(3) (2006)
15. Kerckhoffs A.: La Cryptographie Militaire (Military Cryptography). In: le Journal des Sciences Militaires (Journal of Military Science) IX, 5–38 (1883)
16. Horn B., Szeliski R., Yuille A.: Impossible shaded images. In: IEEE Transactions on PAMI, 15(2), 166–170 (1993)
17. Bimber O., Iwai D., Wetzstein G., Grundhofer A.: The Visual Computing of Projector-Camera Systems. In Eurographics STAR, 23-46 (2007), also CGF, 27(8), 2219-2245 (2008)
18. Grossberg M., Peri H., Nayar S., Belhumeur P.: Making One Object Look Like Another: Controlling Appearance using a Projector-Camera System. In IEEE Conference on Computer Vision and Pattern Recognition, 452-459 (2004)