

Reliefs as Images

Marc Alexa*
TU Berlin & Disney Research, Zurich

Wojciech Matusik†
Disney Research, Zurich

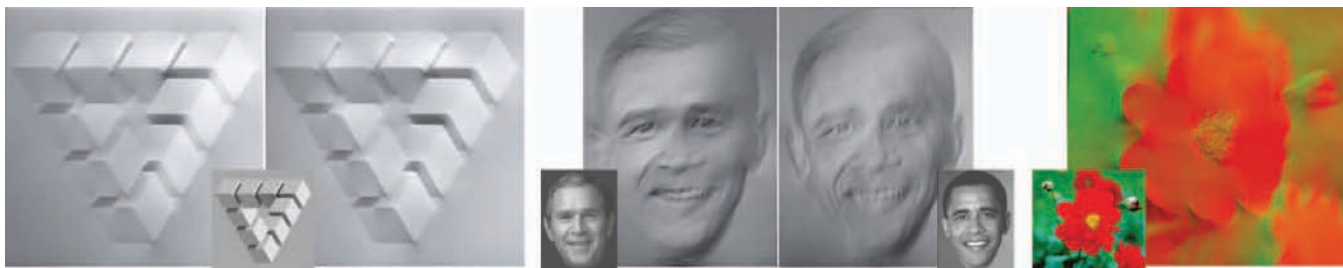


Figure 1: Given arbitrary input images (inset), we compute reliefs that generate the desired images under known directional illumination. The inputs need not depict real surfaces (left), change in light direction can be exploited to depict different images with a single relief surface (middle), and if the light sources are colored, even a white surface can generate a given color image (right).

Abstract

We describe how to create relief surfaces whose diffuse reflection approximates given images under known directional illumination. This allows using any surface with a significant diffuse reflection component as an image display. We propose a discrete model for the area in the relief surface that corresponds to a pixel in the desired image. This model introduces the necessary degrees of freedom to overcome theoretical limitations in shape from shading and practical requirements such as stability of the image under changes in viewing condition and limited overall variation in depth. The discrete surface is determined using an iterative least squares optimization. We show several resulting relief surfaces conveying one image for varying lighting directions as well as two images for two specific lighting directions.

CR Categories: I.3.6 [Computer Graphics]: Picture/Image Generation—Display algorithms; I.3.6 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations;

Keywords: relief, sculpture, geometry generation

1 Introduction

Reliefs are usually created by compressing the depth image of real 3D geometry. This technique has been used for centuries by artists, and has recently been reproduced digitally [Cignoni et al. 1997; Weyrich et al. 2007; Song et al. 2007; Sun et al. 2009]. These

*e-mail:marc.alex@tu-berlin.de

†e-mail:matusik@disneyresearch.com

approaches effectively exploit the bas-relief ambiguity [Belhumeur et al. 1999] in order to predict the image generated by the relief surface under constant uniform illumination and mostly diffuse reflection.

However, the range of images that could be conveyed by a surface is richer than surfaces obtained with bas-relief ambiguity. In this work, we define the following more general problem: given an input image (or a set of images) and the corresponding light directions, we would like to obtain a relief such that its diffuse shading approximates the input image when illuminated by this directional lighting. To our knowledge, this is the first work with this goal in mind.

The theory of shape from shading has considered a similar but different problem: given an observed image that is the result of a single light source illuminating a scene with mostly diffuse reflectance, can we recover the depth image? As part of this theory it has been shown that not every observation can be explained with a constant-albedo, diffuse surface [Horn et al. 1993]: the radiance of a surface element depends on the normal relative to the direction of the light and the viewer, however, the normals of a continuous surface are not independent – the vector field of normals is necessarily *integrable* or *irrotational*.

Without this condition it would be quite simple to solve the problem of generating a desired image as the reflection of a diffuse surface (disregarding interreflection and self-occlusion): if we fix viewing and lighting directions then each pixel value in the image directly defines the angle between surface normals and light direction. In fact, this only exploits one of the two degrees of freedom in each normal. The remaining degree of freedom could be fixed by considering a second light direction. Interestingly, it has been shown that two arbitrary images can indeed be created by a single diffuse surface for two given directional light sources [Chen et al. 2000]. However, this requires varying albedo because, as mentioned, the normals of a surface are constrained.

The main idea and contribution of this work lies in a discrete surface model that effectively alleviates the problem of constrained normals (see section 3): each pixel is associated not only with one but with several elements of the surface. Together, the surface elements provide enough degrees of freedom and the different resulting radiance values are effectively integrated for realistic viewing distance.

We analyze this approach in section 4 and show that it indeed al-

lows generating two images within a mutual gamut under certain idealized assumptions. These assumptions are perfect lighting and viewing conditions and unrealistic machining properties. Asking for one image to be generated for two different light source directions makes the relief image stable against changes in lighting. Furthermore, adding smoothness constraints leads to more flexibility in viewing conditions and easier fabrication.

In section 5 we describe an iterative optimization procedure that finds a solution to the discrete model with the additional constraints. Each step in the iterative optimization requires solving a large but sparse linear system. We also discuss how to control the overall depth of the resulting relief and how to optimize for two or more desired images.

We show several results in section 6. When the same image is used for both light source positions we get relief results that generate stable images for a wide range of light source directions and viewing directions. This is an interesting result, as the images are not necessarily depicting diffuse surfaces under single light source illumination. We also show that it is possible to generate two different images with a single relief surface, which can also be used to generate color reliefs.

We present a rather simple approach, taking into account only direct shading, but no interreflection, sub-surface scattering, and other realistic shading effects. Consequently, we believe that this is only the beginning of using surfaces as displays. More discussion and conclusions are presented in section 7.

2 Related work

Our work is closely related to shape from shading (SFS). Shape from shading is a classic problem in computer vision with the first practical algorithms developed by Horn in early 1970s [1970] (see a survey by Zhang et al. [1999]). Given a grayscale image (or a set of images) of a scene under a point (or directional) light source, SFS algorithms attempt to recover surface shape and light source information. In the most basic version, the surface is assumed to be Lambertian and the light source location is known. Thus, the problem settings for this case and our problem setting appear to be quite similar. However, there are some important differences. In our case, we deal with arbitrary input images. Our input might not be a photograph or it might correspond to a scene illuminated by many light sources, a scene with non-uniform albedo or BRDF and depth discontinuities. Thus, we do not attempt to recover the actual scene geometry but merely produce a relief that can induce the desired image. Moreover, an important objective for us is to keep the depth of the relief small as it is difficult to fabricate surfaces with large depth variation. Furthermore, we also show how to embed two images into one relief.

Another set of algorithms related to ours are methods that produce bas-reliefs [Cignoni et al. 1997; Weyrich et al. 2007; Song et al. 2007; Sun et al. 2009]. In this problem, one starts with in a depth-map of a three-dimensional model from a particular viewpoint and the goal is to compress this depth to produce a relief with small depth variations but which still depicts this model and its shading faithfully. In bas-reliefs the illumination is assumed to be uniform, while in our case we optimize for a range of light directions. Moreover, the key difference is that the input to our system is only an image. We believe that using images as input is much more practical in most cases.

Finally, our work is also related to Shadow Art by Mitra and Pauly [2009]. They present an optimization process that uses as input up to three binary images and computes a 3D volume that casts shadows that best approximate the input stencils.

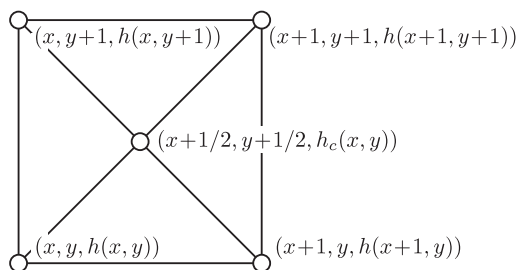


Figure 2: The discrete representation of a pixel in the relief and notation.

3 Discrete relief model of a pixel

We will represent the relief by a discrete surface, fixing the vertex positions in the plane and only adjusting the heights (i.e. a discrete height field). A simple counting argument shows that a regular grid of vertices, associating one quadrilateral element to each pixel, seriously limits the space of images: asymptotically, there is one degree of freedom per element. This is independent of whether we consider non-planar quadrilateral elements or triangulate each element to form a piecewise planar surface. The limitation is nicely illustrated in the survey by Zhang et al. [1999][Fig. 13–18] by applying several shape from shading algorithms to images that were not photographs of diffuse surfaces. Horn et al. [1993] have analyzed the problem theoretically and give canonical images that cannot be the result of shading a smooth surface (see Fig. 3 left).

We overcome this limitation by adding an additional degree of freedom: each element of the relief surface corresponding to a pixel in the images is modeled as four triangles forming a square in the parameter domain (see Fig. 2). The four triangles will have different normals and, thus, different foreshortening. When viewed from a distance a human observer will perceive the aggregate radiance of all four facets. Note that the one additional vertex in the center of each element adds exactly the one necessary degree of freedom per element – in this model, each element corresponding to a pixel in the image offers (asymptotically) two degrees of freedom.

Assume for the moment that two light source directions $\mathbf{l}^0, \mathbf{l}^1$ are fixed, and two discrete gray level images are given $I^0, I^1 \in \mathbb{R}^{m \times n}$. We further assume the viewer is distant relative to the size of the relief, i.e. in direction $\mathbf{v} = (0, 0, 1)$. The goal is to create a discrete relief surface that reproduces each pixel’s intensity as the integrated radiance of the corresponding 4 triangles.

Derivation of radiance per pixel. Let the vertices around each pixel be denoted as

$$\mathbf{p}(x, y) = (x, y, h(x, y)) \quad (1)$$

and the center vertex surrounded (counterclockwise) by $\mathbf{p}(x, y), \mathbf{p}(x+1, y), \mathbf{p}(x+1, y+1), \mathbf{p}(x, y+1)$ be

$$\mathbf{p}_c(x, y) = \left(x + \frac{1}{2}, y + \frac{1}{2}, h_c(x, y) \right). \quad (2)$$

We want to compute the radiance of a surface element as viewed from the $\mathbf{v} = (0, 0, 1)$ direction. First note that the projected area of all four triangles is $1/4$. We denote the cross products of edges



Figure 3: Creation of reliefs for the canonical “impossible” image (left) from [Horn et al. 1993]: using pyramids we could create a qualitatively perfect reconstruction of the image (i.e. the gradient images are almost identical), albeit at the expense of pronounced pyramid, which cause extreme dependence on perfect viewing and illumination conditions (center). We suggest to optimize the heights of vertices and impose smoothness constraints and arrive at the reconstruction shown on the right (which is worse based on comparing the image gradients). The insets show a perspective view over the surface, illustrating the clearly visible pyramids in the center image and no visible variation in heights in the right image.

incident on the center vertex as

$$\begin{aligned}\tilde{\mathbf{n}}_0(x, y) &= (\mathbf{p}_c(x, y) - \mathbf{p}(x, y)) \times (\mathbf{p}_c(x, y) - \mathbf{p}(x+1, y)) \\ \tilde{\mathbf{n}}_1(x, y) &= (\mathbf{p}_c(x, y) - \mathbf{p}(x+1, y)) \times (\mathbf{p}_c(x, y) - \mathbf{p}(x+1, y+1)) \\ \tilde{\mathbf{n}}_2(x, y) &= (\mathbf{p}_c(x, y) - \mathbf{p}(x+1, y+1)) \times (\mathbf{p}_c(x, y) - \mathbf{p}(x, y+1)) \\ \tilde{\mathbf{n}}_3(x, y) &= (\mathbf{p}_c(x, y) - \mathbf{p}(x, y+1)) \times (\mathbf{p}_c(x, y) - \mathbf{p}(x, y))\end{aligned}\quad (3)$$

and then their lengths as

$$n_i(x, y) = \|\tilde{\mathbf{n}}_i(x, y)\|, \quad i \in \{1, 2, 3, 4\}.\quad (4)$$

With this notation the reflected radiance of the surface element $L(x, y)$ illuminated by a light source \mathbf{l} is

$$L(x, y) = \sum_{i=1}^4 \frac{1}{4} \cdot \frac{\tilde{\mathbf{n}}_i(x, y)^\top}{n_i(x, y)} \mathbf{l} = \frac{1}{4} \left(\sum_{i=1}^4 \frac{\tilde{\mathbf{n}}_i(x, y)}{n_i(x, y)} \right)^\top \mathbf{l}.\quad (5)$$

It is important for the optimization procedure we describe later that $\tilde{\mathbf{n}}_i$ is a linear function of the heights $\{h_i\}, h_c$ – we have deliberately put the non-linear part of the radiance function into the lengths n_i . For ease of re-implementation, we write out the linear equations for the radiance in terms of the variables $\{h_i\}$ and h_c relative to a light direction $\mathbf{l} = (l_x, l_y, l_z)$:

$$\begin{pmatrix} \frac{1}{2}(l_x + l_y) \left(\frac{1}{n_1} + \frac{1}{n_4} \right) \\ \frac{1}{2}(-l_x + l_y) \left(\frac{1}{n_1} + \frac{1}{n_2} \right) \\ \frac{1}{2}(-l_x - l_y) \left(\frac{1}{n_2} + \frac{1}{n_3} \right) \\ \frac{1}{2}(l_x - l_y) \left(\frac{1}{n_3} + \frac{1}{n_4} \right) \\ l_x \left(\frac{1}{n_2} - \frac{1}{n_4} \right) + l_y \left(\frac{1}{n_1} - \frac{1}{n_3} \right) \end{pmatrix}^\top \begin{pmatrix} h(x, y) \\ h(x+1, y) \\ h(x, y+1) \\ h_c(x, y) \end{pmatrix} + l_z \sum_{i=1}^4 \frac{1}{n_i} = 4L(x, y)\quad (6)$$

In Fig. 3 we illustrates how much flexibility can be gained from using pyramid shaped elements. As the input image we use a canonical impossible image example by Horn et al. [1993]. Then we show what was possible if we used arbitrary pyramids for the reconstruction. Unfortunately, the resulting variation in height makes the result unstable under even slight changes in viewing direction, as many of the triangles are at grazing angles with the light sources. In addition, it would be difficult to machine the surface with common tools. Because of these problems we impose additional smoothness constraints (see the following sections).

4 Analysis and consequences of practical requirements

In this section we analyze what radiance values can be generated with our geometric model for a pixel. The additional degree of freedom in each pixel suffices to control the radiance per pixel, independently of other pixels. With two degrees of freedom per pixel it could be possible to select independent radiance values for two different light source directions. However, we show that the gamut of radiances for an individual pixel is limited, even under idealized assumptions. In other words, if the images are supposed to differ for different illumination their dynamic range will be limited; a large dynamic range can only be achieved for similar or identical images.

Then we analyze consequences for real life conditions. Most importantly, the viewing position cannot be controlled and, thus, the image cannot be too sensitive against changes in viewing direction. We show that the derivatives of the height field are directly related to the stability of the image under changes in viewing direction. This will lead to smoothness terms in the surface optimization.

Our discussion complements the analysis in the shape from shading literature [Zhang et al. 1999]. We focus on the properties of the output regardless of the input and, in particular, input that cannot be the result of shading a real surface.

4.1 Radiance gamut for two light sources

Assume two fixed light source directions \mathbf{l}^0 and \mathbf{l}^1 . Moreover, the direction towards the viewer is $\mathbf{v} = (0, 0, 1)$ and the relief surface is in the xy -plane. Our goal is to create a surface element that generates two different radiance values when it is illuminated by two different directional light sources. We restrict the analysis to a planar surface patch (i.e. a triangle) as the radiance for the pyramid is a convex combination of the radiance for the 4 triangles. The problem of generating two arbitrary radiance values for the two light directions is equivalent to finding a solution for a surface normal \mathbf{n} that satisfies the following two equations:

$$M_0 = \mathbf{n}^\top \mathbf{l}^0 \quad M_1 = \mathbf{n}^\top \mathbf{l}^1.\quad (7)$$

Geometrically, the pairs M_0, \mathbf{l}^0 and M_1, \mathbf{l}^1 define two cones of revolution with axes \mathbf{l}^0 and \mathbf{l}^1 . The intersection of these cones results in 0, 1, or 2 surface normals. In practice, we only consider the parts of the cones that have positive z -component, as any solution \mathbf{n} needs to point towards the viewer. Under these conditions the subspace of possible radiance values in both images $M_0 \times M_1 = [0, 1]^2$ that permits at least one solution for each surface normal \mathbf{n} depends on the directions of the light sources. However, we observe that it is necessarily a strict subset of $[0, 1]^2$. For example, if one pixel requires $M_0 = M_1 = 1$ this would lead to $\mathbf{n} = \mathbf{l}^0 = \mathbf{l}^1$. In other words, there is no configuration that would allow reproducing any two arbitrary images. Fig. 4 illustrates several achievable radiance gamuts for a few different light configurations. Note that the radiance gamut depends only on the angle between the light sources. Intuitively, achieving pairs of bright values requires the light sources to have similar direction. This comes at the expense of limiting the maximum difference between the corresponding radiance values in both images.

4.2 Sensitivity to changes in viewing direction

We also would like be able to view the generated reliefs from arbitrary positions. This violates our initial assumption that the viewer is at $\mathbf{v} = (0, 0, 1)$. As we relax the assumption on the viewing direction, the observed radiance at a pixel will vary as the viewer

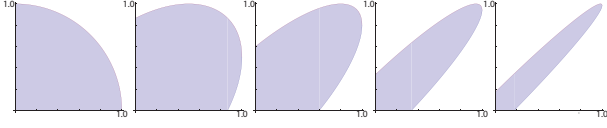


Figure 4: Achievable radiance gamuts for light directions $\mathbf{l}_0 = (1 + h^2)^{-1/2}(1, 0, h)^T$ and $\mathbf{l}_1 = (1 + h^2)^{-1/2}(0, 1, h)^T \sqrt{1 + h^2}$ for values $h = \{0, 1, 2, 4, 8\}$.

moves due to changes in projected areas of the facets. More precisely, while the reflected radiance of each triangle is of course independent of the viewing direction, its projected area changes with the viewing direction; thus, if neighboring triangles within a pixel have different normals, their combined radiance will now be view-dependent. Consequently, we wish to analyze how radiance values change as the observer departs from the ideal viewing direction.

Consider the triangle $(0, 0, h_0), (1, 1, h_c), (2, 0, h_1)$ (we have scaled the coordinates by a factor of two for convenience of notation). The cross product of two edge vectors $(h_0 - h_1, h_0 + h_1 - 2h_c, 2)^T$ is orthogonal to the plane containing the triangle and has length proportional to the area of the triangle. The projected area of the triangle along the direction $\mathbf{v} = (v_x, v_y, v_z)$ is proportional to the scalar product

$$v_x(h_0 - h_1) + v_y(h_0 + h_1 - 2h_c) + 2v_z. \quad (8)$$

Thus, a small deviation along v_x behaves as $h_0 - h_1$ and along v_y as $h_0 + h_1 - 2h_c$. This means, the change in area due to changes in viewing position away from the assumed direction orthogonal to the surface depends on the difference in height values (or the gradient of the height function). For small gradients the radiance is constant under small changes in viewing positions, while for larger gradients the radiance changes linearly with the viewing position.

As mentioned, this change in area is only relevant if it occurs at different rates for neighboring triangles. If all four triangles in a pixel have similar slope, the change in area will be similar and there will be no perceivable change in radiance for this pixel. On the other hand, if the slopes are different, the areas might change differently and the combined radiance would also change. Consequently, it is important that second order differences within a pixel are small; or, put simply, that each pixel is rather flat.

As we can see, the geometric pixel model we are proposing overcomes the limited set of images that can be represented by diffuse surfaces at the expense of sensitivity to viewing direction. The optimization procedure described in the next section allows trading off between these goals.

5 Optimization

The overall goal of the optimization process is to compute a surface model whose diffuse shading reproduces the desired image or images for given directional light sources. In this optimization process, we minimize the squared difference of radiance gradients and the corresponding input image gradients. We also add smoothness and damping terms based on the analysis described in section 4. Next, we derive each of the error terms. Then, we describe in more detail the optimization procedure.

Radiance term. As discussed in the previous section the dynamic range of shading produced by relief surfaces is quite limited. Therefore, it is desirable to tone map the input images as a preprocessing step. We use the image gradient compression method similar to approaches by Fattal et al. [2002] and Weyrich et al. [2007]. We

denote the compressed image gradients as:

$$\begin{aligned} D_x^b(x, y) &= C \left(I^b(x+1, y) - I^b(x, y) \right) \\ D_y^b(x, y) &= C \left(I^b(x, y+1) - I^b(x, y) \right) \end{aligned} \quad (9)$$

where C is a function that compresses the value of its argument. In our experiments we have used

$$C(a) = \text{sgn}(a) \cdot \frac{1}{\alpha_d} \log(1 + \alpha_d |a|), \quad (10)$$

similar to the depth compression function used by Weyrich et al. [2007].

The gradients of radiance values for the relief illuminated by the light source l^b are denoted as $L_x^b(x, y)$ and $L_y^b(x, y)$. They can be derived simply by taking finite differences of Equation 5 among neighboring pixels:

$$\begin{aligned} L_x^b(x, y) &= L^b(x+1) - L^b(x, y) \\ L_y^b(x, y) &= L^b(y+1) - L^b(x, y). \end{aligned} \quad (11)$$

Given these definitions, we minimize the squared differences between radiance gradients and corresponding compressed image gradients:

$$\begin{aligned} E_x^b(x, y) &= w_x^b(x, y) \left(L_x^b - D_x^b(x, y) \right)^2 \\ E_y^b(x, y) &= w_y^b(x, y) \left(L_y^b - D_y^b(x, y) \right)^2. \end{aligned} \quad (12)$$

Note that we use weights $w_{\{x,y\}}^b(x, y)$ that depend on the light source direction, the gradient direction, and the position. We will discuss the choice of these weights later. The global energy for fitting the gradients of the input images is then:

$$E_g^b = \sum_{x=1}^m \sum_{y=1}^n E_x^b(x, y) + \sum_{x=1}^m \sum_{y=1}^n E_y^b(x, y) \quad (13)$$

Smoothness and height control. As discussed in section 4.2 it is also important that each pixel is close to flat. This can be achieved by minimizing the second order differences within a pixel. For center vertices, this term is expressed using a sum of their squared Laplacians:

$$\begin{aligned} E_c = \sum_{x=1}^m \sum_{y=1}^n & \left(h(x, y) + h(x+1, y) + \right. \\ & \left. h(x+1, y+1) + h(x, y+1) \right. \\ & \left. 4h_c(x, y) \right)^2 \end{aligned} \quad (14)$$

Intuitively minimizing this term pulls each center vertex towards the mean height of its neighbors. However, moving the center vertex to the mean height is not enough to keep the geometry of a pixel planar. It is also necessary to consider the second order smoothness of corner vertices of each pixel. This term for all pixels can be expressed as:

$$\begin{aligned} E_p = \sum_{x=1}^m \sum_{y=1}^n & \left(h(x, y) - h(x+1, y) + \right. \\ & \left. h(x+1, y+1) - h(x, y+1) \right)^2. \end{aligned} \quad (15)$$

Lastly, we penalize large height values. We add another term that ensures the resulting heights are close to desired heights $h^*(x, y)$:

$$E_h = \sum_{x=1}^m \sum_{y=1}^n \left(h(x, y) - h^*(x, y) \right)^2. \quad (16)$$

We discuss the choice of the desired heights $h^*(x, y)$ later. We also note that a similar energy for center vertices is not necessary, as the height of these vertices is locally constrained.

Minimizing the energy. The overall energy including the weights for the regularization terms can be expressed as follows:

$$E = E_g^0 + E_g^1 + w_c E_c + w_p E_p + w_h E_h. \quad (17)$$

Our optimization relies on the assumption that the overall height of the desired relief should be small as discussed in section 4. The main difficulty in optimizing the energy terms is due to the non-linearity of $L^b(x, y)$. This expression is non-linear because of its components n_i that denote the lengths of the cross products $\tilde{\mathbf{n}}_i$ corresponding to the areas of the triangles forming pixels. However, following the discussion on sensitivity to viewing direction we explicitly keep the change in these components small by means of Equations 14 and 15. Exploiting this damping, we model the $\{n_i\}$ as constants in each step of an iterative optimization process. This means that the gradient of (12) becomes linear.

The choice of $h^*(x, y)$ depends on the application scenario. In some cases the desired heights are provided as discussed in section 6. If they are not given then the heights are damped only as they get too large. In practice, for small heights we set $h^*(x, y)$ to $h(x, y)$, which allows the heights to vary over the iterations. For large heights, we reduce their values after each iteration. In particular, we have found that using a compression function similar to (10) and repeatedly setting

$$h^*(x, y) = \alpha_h \log(1 + \alpha_h h(x, y)) \quad (18)$$

works well in practice.

The complete optimization method can be described as follows. In each step we set the linearized gradient of the energy to zero. This corresponds to solving a sparse linear system for the unknown heights. Then we update the constants $\{n_i\}$ and $\{h^*\}$ and repeat the steps. This procedure typically converges in a few iterations (5-10) when the constraint enforcing small height variation permits a solution. If no solution with small height variation is possible then the procedure diverges quickly.

Possible generalizations The iterative nature of the optimization process allows us to generalize the input illumination setting. For example, point (or even area) light sources can be modeled by approximating them with a single or several directional light sources in each iteration. Moreover, the process can be easily adapted to optimize for light directions or positions. We have experimented with this extension of our method and we have found that it works well in practice.

6 Results and applications

We use the proposed optimization method to create example surfaces that convey various input images. In all examples we optimize for two or three directional light sources which lie within a 180° cone centered at the main surface normal. We observe that in order to obtain the best results, the average value of a given input image should match the radiance of a relief plane under the corresponding light source. This can be achieved either by adjusting

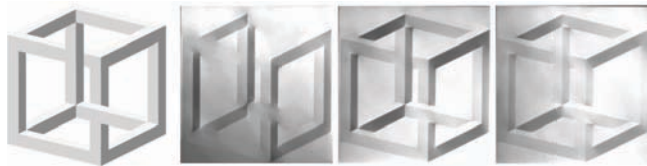


Figure 5: The desired image, and the resulting machined relief surface for different lighting directions.

the light source intensity or by preprocessing the input images. We have also determined that the optimization parameters introduced in the last section can be mostly set to default values. The smoothness terms w_c , w_p are set to 0.1 for all results and the gradient compression factor α is set to one (except where noted). The only parameter that needs to be adjusted is the height control w_h .

In our examples we use input images with resolutions between 300 by 300 pixels and 600 by 600 pixels. The resolution of the resulting reliefs is matched or exceeds the resolution of the input images. The computed surfaces are stored as STL files, which can then be used to fabricate the physical surfaces. We manufacture these surfaces using two different fabrication methods. In the first method, we use a Roland EGX-600 computer controlled engraver. This engraving machine is very precise (0.01 mm). Its working volume is $610 \times 407 \times 42.5$ mm. However, in practice, the maximum height of a relief generated using this engraver is about 5 mm. Our example reliefs are cut out of acrylic and then they are coated with a white diffuse paint. For a second method, we use a high-precision 3D printer – Connex350 by OBJET. The maximum resolution of the printer is 600dpi and the working volume is $342 \times 342 \times 200$ mm. This 3D printer lets us generate reliefs with larger depth range. As the base printing material is slightly translucent we also coat all the example surfaces with a white diffuse paint. All figures in this paper include photographs of the relief surfaces produced using one of these methods.

Single grayscale image. When generating a relief that depicts a single grayscale image under different light directions we optimize using the same input image for all desired light directions. The resulting relief image is usually slightly different for different directional illuminations. However, the image can be clearly perceived even for light directions that have not been part of the optimization. The case of depicting a single grayscale image is less constrained and therefore we can afford to make the surfaces relatively flat – we use $w_h = 1$.

As an interesting example, we have manufactured surfaces that depict “impossible” geometry such as the impossible triangle (see Fig. 1) or the famous impossible cube (see Fig. 5). For the cube we show several different lighting directions. To show how insensitive the reliefs are to the actual illumination we took a photograph of the relief in Fig. 6 under very general lighting conditions, i.e. daylight that has a dominant direction because it comes through a window.

Pairs of images. We have also created several reliefs that depict two different images when illuminated from two distinct light directions. We observe that high contrast edges that are present in the input images are visible in the relief surface regardless of the light direction. While unfortunate, this cannot be avoided when the output surface is reasonably smooth (see our gamut analysis in section 4.2 and also Horn et al.[1993]). This is because a high contrast edge in the image corresponds to a line of high curvature in the fabricated surface. This edge is going to be visible unless the di-



Figure 6: A relief created from the image in the top row. The illumination for capturing the image is created by simply putting the relief close to a window (middle row), showing that the images created by the reliefs are not very sensitive to the particular illumination setting.



Figure 7: Pairs of images taken under different illumination conditions, resulting optimized simulated pair and photographs of the machined surface for roughly equivalent light directions.

rection of this curvature maximum is aligned with the projection of the light direction. This requirement is violated by any edge that is curved almost everywhere. Therefore, input image pairs that can be reliably reproduced have contrast edges in similar places or in regions where the other image is stochastic.

Perhaps the most obvious input to our method are two photographs of the same scene taken under different illumination conditions. We

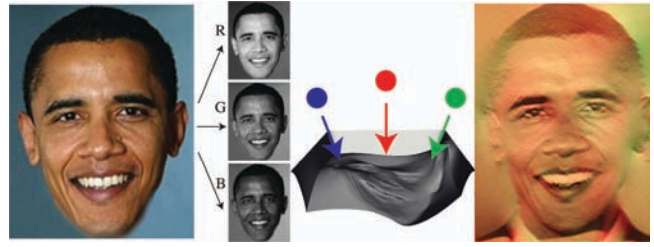


Figure 8: A relief created to generate a color image under illumination by a red, a green, and a blue light source. Due to the large differences in desired radiance values for some regions it is hard to avoid a significant depth of the relief.

show such examples in Fig. 7. Note that most parts of the resulting relief are not good approximations of the underlying geometry of the scene. However, even for a moving light source the relief behaves realistically.

We have also tested the reproduction on sufficiently different input images. The main artifact in these examples are the visible high contrast edges for most light directions. In order to reduce this problem we can align the input images. For example, in Fig. 1 we show a result for which we warp two different face images such that the eyes, nose, and mouths coincide. In the case of two input images, different radiance values for the same surface location leave less room for the choice of normals. This means for a reasonable reproduction we cannot compress the height as much as in the single image case. For these examples, we reduce the value of w_h to 0.1.

Color image reliefs. As another application we consider the case when the light sources have different colors. Since our method allows depicting two or more similar input images we attempt to reproduce colored images as well. Given the colors of the light sources, we project a color image into these primaries yielding a set of input images. These images are used as input to our optimization process to obtain a surface which approximates the color image when illuminated by this colored lighting. We believe that these reliefs are the first to reproduce color images on white diffuse surfaces.

The example in Fig. 1 uses two primaries, a red and a green light. We have also tried decomposition into RGB and using three differently colored light sources (see Fig. 8). However, we have found that full color reproduction is usually very limited because the desired radiance for different channels can be very different. This leads to necessarily large inclination of the surface. We allow this behavior by reducing the height control parameter w_h to 0.001. We believe that it is possible to improve these results by also optimizing for light source directions and color.

Geometry / image pairs. As the last application, we show that we can combine traditional reliefs (or bas-reliefs) with additional desired images. We have modified the RTSC software courtesy of DeCarlo and Rusinkiewicz [DeCarlo et al. 2003] to output the shade images and the corresponding depth maps. We use one of the depth maps to provide values for $h^*(x, y)$. The input images are used to set additional constraints. Fig. 9 shows the result of using mean curvature shading and black contours as the input images. The resulting relief shows significantly more detail compared to the plain relief since it also reproduces the desired input images. Note that we are not specifically aiming at the creation of bas-reliefs such as the prior work [Weyrich et al. 2007; Song et al. 2007; Sun et al. 2009].

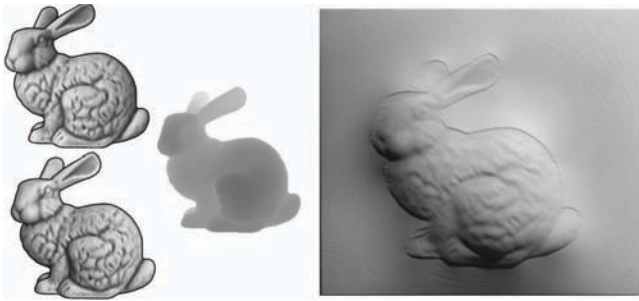


Figure 9: This relief has been created from images produced by RTSC [DeCarlo et al. 2003] and a corresponding depth image to set the values of $h^*(x, y)$. The mean curvature shading used in the images introduces additional detail into the relief that is not a part of the shading due to the depth image.

7 Conclusions

We have demonstrated that it is feasible to automatically generate relief surfaces that reproduce desired images under directional illumination. Our experiments show that it is important to include requirements such as invariance due to changes in viewing and illumination direction in the optimization approach. The reliefs resulting from our optimization clearly show the desired images despite imperfect fabrication, non-Lambertian reflectance properties of base materials, and a wide range of viewing and lighting conditions.

Limitations: Generating reliefs for a single image, visible under different illuminations works best. However, designing reliefs such that they produce different images for different light directions suffers from significant “cross-talk” between high contrast edges. While these edges typically align nicely in the color setting, the necessary large differences in image values for the corresponding pixels lead to either large gradients in the relief making it hard to fabricate and leading to less stable viewing conditions. We believe that color reliefs could be improved significantly by extracting two dominant primaries from the input image (for example, using an optimization approach similar Power et al. [1996]) and then projecting the image into the new color space. This, however, would make the practical setup of the light sources more difficult. Furthermore, our assumption that the reflectance of underlying material is Lambertian is an approximation, as reflectance of real surfaces may have a significant non-Lambertian component. However, in practice, the images are still clearly visible even on glossy surfaces. Our current approach also does not take into account interreflections between different parts of the relief surface. Since our reliefs are typically quite flat this effect is not very noticeable.

Future directions: The limitations mentioned above are obvious directions for future research. One may use arbitrary BRDFs, consider subsurface scattering, or interreflections during the optimization of relief surfaces. Furthermore, given that the surfaces are fabricated with computerized tools, it is conceivable that their albedo can be controlled as well. This would make the optimization process less constrained and it would very likely lead to better visual results. Finally, it is possible to optimize reliefs for point light sources instead of directional lights. This would extend the applicability of our method as many real world lights can be approximated using a point light source model. Furthermore, using point light sources might also add an additional degree of freedom and allow generating more than two images with a single relief surface.

Acknowledgements

The project has been inspired by discussions with Bill Freeman and Szymon Rusinkiewicz on manipulating shading to reproduce images. We also thank Paul Debevec and his group at USC Institute for Creative Technologies for providing images (Fig. 7) from the reflectance field demo. The bunny model (Fig. 9) is from the Stanford 3D Scanning Repository, the red flower image (Fig. 1) from photos.a-vsp.com, and all other source images are retrieved from wikimedia.

References

- BELHUMEUR, P. N., KRIEGMAN, D. J., AND YUILLE, A. L. 1999. The bas-relief ambiguity. *Int. J. Comput. Vision* 35 (November), 33–44.
- CHEN, H., BELHUMEUR, P., AND JACOBS, D. 2000. In search of illumination invariants. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-00)*, IEEE, Los Alamitos, 254–261.
- CIGNONI, P., MONTANI, C., AND SCOPIGNO, R. 1997. Computer-assisted generation of bas- and high-reliefs. *J. Graph. Tools* 2 (December), 15–28.
- DECARLO, D., FINKELSTEIN, A., RUSINKIEWICZ, S., AND SANTELLA, A. 2003. Suggestive contours for conveying shape. *ACM Trans. Graph.* 22 (July), 848–855.
- FATTAL, R., LISCHINSKI, D., AND WERMAN, M. 2002. Gradient domain high dynamic range compression. *ACM Trans. Graph.* 21, 3, 249–256.
- HORN, B. K. P., SZELISKI, R. S., AND YUILLE, A. L. 1993. Impossible shaded images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15, 2, 166–170.
- HORN, B. K. P. 1970. *Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View*. PhD thesis, Massachusetts Institute of Technology.
- MITRA, N. J., AND PAULY, M. 2009. Shadow art. *ACM Transactions on Graphics* 28, 5, 1–7.
- POWER, J. L., WEST, B. S., STOLLNITZ, E. J., AND SALESIN, D. H. 1996. Reproducing color images as duotones. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 237–248.
- SONG, W., BELYAEV, A., AND SEIDEL, H.-P. 2007. Automatic generation of bas-reliefs from 3d shapes. In *Proceedings of the IEEE International Conference on Shape Modeling and Applications 2007*, IEEE Computer Society, Washington, DC, USA, 211–214.
- SUN, X., ROSIN, P. L., MARTIN, R. R., AND LANGBEIN, F. C. 2009. Bas-relief generation using adaptive histogram equalization. *IEEE Transactions on Visualization and Computer Graphics* 15 (July), 642–653.
- WEYRICH, T., DENG, J., BARNES, C., RUSINKIEWICZ, S., AND FINKELSTEIN, A. 2007. Digital bas-relief from 3d scenes. *ACM Trans. Graph.* 26, 3, 32.
- ZHANG, R., TSAI, P.-S., CRYER, J. E., AND SHAH, M. 1999. Shape from shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21, 8, 690–706.