

# Interactive Normal Reconstruction from a Single Image

Tai-Pang Wu<sup>1</sup>

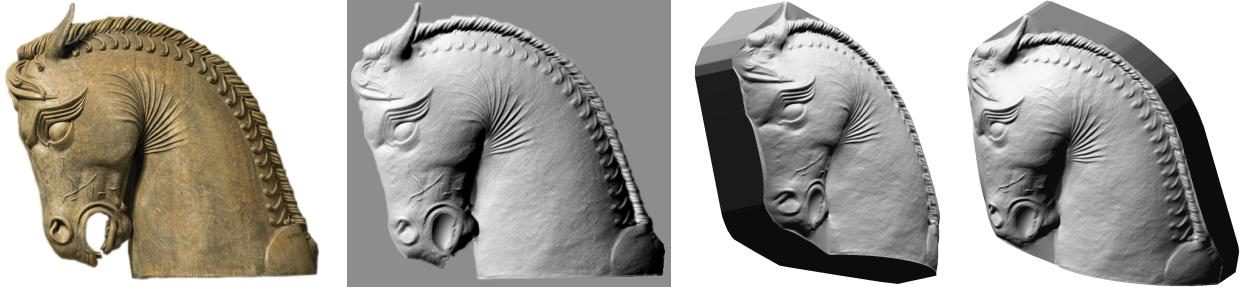
Jian Sun<sup>1</sup>

Chi-Keung Tang<sup>2</sup>

Heung-Yeung Shum<sup>1</sup>

<sup>1</sup>Microsoft Research Asia

<sup>2</sup>The Hong Kong University of Science and Technology



**Figure 1:** Horse. From left to right: input image, surface normals generated by our interactive system rendered using Lambertian shading, and two views of the reconstructed surface.

## Abstract

We present an interactive system for reconstructing surface normals from a single image. Our approach has two complementary contributions. First, we introduce a novel shape-from-shading algorithm (SfS) that produces faithful normal reconstruction for local image region (high-frequency component), but it fails to faithfully recover the overall global structure (low-frequency component). Our second contribution consists of an approach that corrects low-frequency error using a simple markup procedure. This approach, aptly called *rotation palette*, allows the user to specify large scale corrections of surface normals by drawing simple stroke correspondences between the normal map and a sphere image which represents rotation directions. Combining these two approaches, we can produce high-quality surfaces quickly from single images.

## 1 Introduction

Normals are important for many graphics applications ranging from re-lighting, texture mapping [Fang and Hart 2004], to material editing [Khan et al. 2006] and surface decoration. In this paper, we present an interactive system to recover surface normals from single images. Recovering normals from single images has unique advantages on model acquisition: innumerable photos have been and will be captured, and these photos are effortlessly available from the word wide web.

Automatic normal estimation methods, i.e., Shape-from-Shading (SfS) [Horn and Brooks 1989], compute surface normal from the shading information in the image. It is well known, however, that automatic SfS algorithms are error-prone due to the ill-posedness of the SfS problem. As will be reviewed in the related work section,

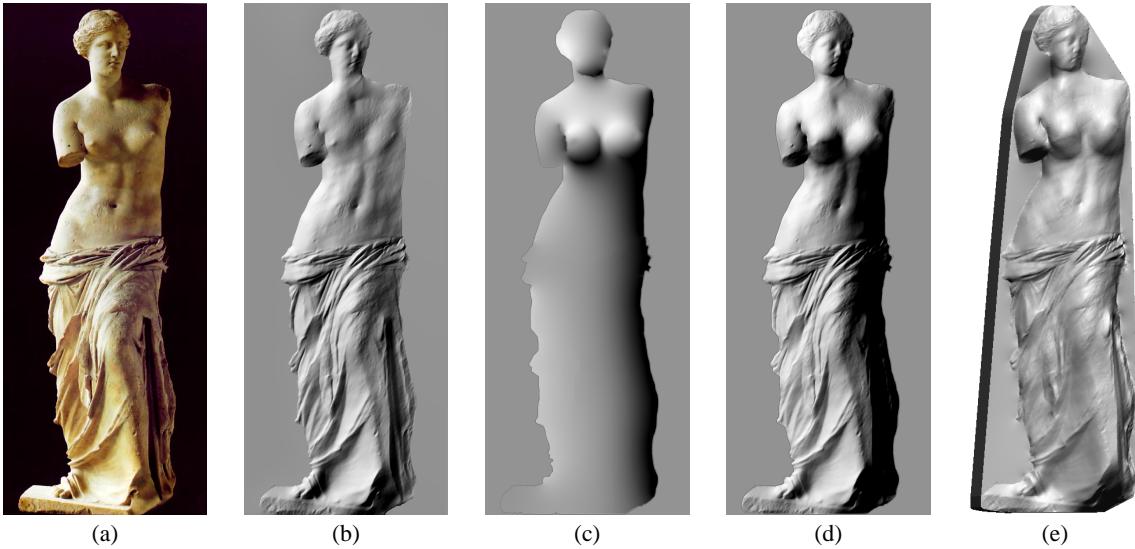
restrictions and assumptions in SfS algorithms make them insufficient to produce high quality surface normals. On the other hand, existing interactive methods [Zhang et al. 2001; Zeng et al. 2005; Prasad et al. 2006] can improve automatic approaches by having the user specify absolute surface positions or absolute surface normals as constraints. For an image with complex geometry (e.g. Figure 1), a large number of constraints will need to be specified.

Our single image approach is based on the observation that low-frequency surface editing is easy for the user, while the automatic SfS algorithm is good at estimating mid/high-frequency surface details. This leads us to develop an interactive approach that allows the reconstruction of high-quality visible surface models of complex image objects (see Figure 1), by combining human interaction and geometry information in the image.

Specifically, our approach consists of two steps. First, we propose a SfS algorithm which globally distributes reconstruction errors via surface reconstruction with the osculating arc constraint, so that errors in normal directions mainly lie in the low frequency component. Our algorithm works best for surface made of homogeneous material. Second, instead of requiring absolute surface positions or absolute surface normals, the user in our system operates on *relative normals* to fix the low-frequency bias of the SfS result. The relative normals can be intuitively manipulated by drawing strokes on a sphere. More importantly, only very sparse markup is needed, and such markup can be conveniently done on a 2D interface. The sparse normal manipulations are propagated on the whole surface by an optimization method. In essence, the relative normal specification is *what* needs to be specified, while the markup approach using a directional sphere is *how* to specify it. Our idea of using relative normals is inspired by successful interactive mesh editing systems [Sorkine et al. 2004; Yu et al. 2004] which perform calculation in the gradient domain, where both methods operate on relative quantities. Our markup idea is inspired by the recently proposed normal markup approach [Wu et al. 2007] that relied on familiar shapes.

## 2 Related Work and Motivation

Given the vast number of high-resolution digital photos available, we expect an increasing demand from general users on interactive modeling from single images. Single view metrology [Criminisi et al. 2000] makes use of vanishing points and invariance in projective geometry to reconstruct a rectilinear scene. The user



**Figure 2:** Venus: (a) input image, (b) the estimated initial normal map, (c) the dense transformation map  $\mathbf{V}$  propagated from sparse user markup on rotation palette, (d) the final normal map, obtained by applying  $\mathbf{V}$  in (c) to the initial normal map in (b), and (e) the reconstructed surface.

of “tour in picture” [Horry et al. 1997] reconstructs a rectilinear scene interactively by identifying vanishing points and fitting a spidery mesh. Automatic photo pop-up [Hoiem et al. 2005] creates from a single photo a 3D model made up of planar texture-mapped primitives. In [Prasad et al. 2006], a single-view reconstruction algorithm for smooth and curved surfaces was proposed. The “SmoothSketch” [Karpenko and Hughes 2006] introduced a system for estimating 3D smooth shapes from visible-contour sketches. A shading-based surface editing approach was proposed in [Gingold and Zorin 2008], which converts user drawing into geometric constraints to alter surface shape. A 3D model, however, is still needed, and the method is more suitable for small-scale editing. Overall, these modeling methods are not designed for modeling complex surfaces with intricate details from single images.

Complex objects have a lot of high-frequency surface details, so thoroughly relying on user specification is impractical. Ideally, they should be automatically recovered from the color/shading of the image. This is the classical SfS problem, which was first formulated by Horn and Brooks [1989]. The typical output consists of noisy surface gradients (normals). Although methods were available to enforce the integrability constraint to reduce the effect of noise (e.g., [Frankot and Chellappa 1988; Kovesi 2005]), details are smoothed out in the resulting surfaces. From a theoretical perspective, [Belhumeur et al. 1999] proved that even under the simplified Lambertian assumption, it is impossible to obtain an exact surface from a single image with unknown light sources and albedoes, a typical input in single view modeling. Only a family of surfaces can be obtained, which are described by the generalized bas-relief transformation.

To limit the bas-relief ambiguity to a binary scale, all the following conditions have to be satisfied [Belhumeur et al. 1999]: i) lighting directions are known; ii) at least three images are available; iii) constant light source intensity/constant albedo; and iv) the surface is Lambertian. Potetz [2007] proposed an efficient belief propagation algorithm, and showed decent SfS results where the surfaces are produced with a probabilistic shape prior. However, the problem associated with the bas-relief ambiguity still remains unsolved. To make the problem well-posed, [Prados and Faugeras 2005] considered a more realistic lighting model, where the attenuation of a

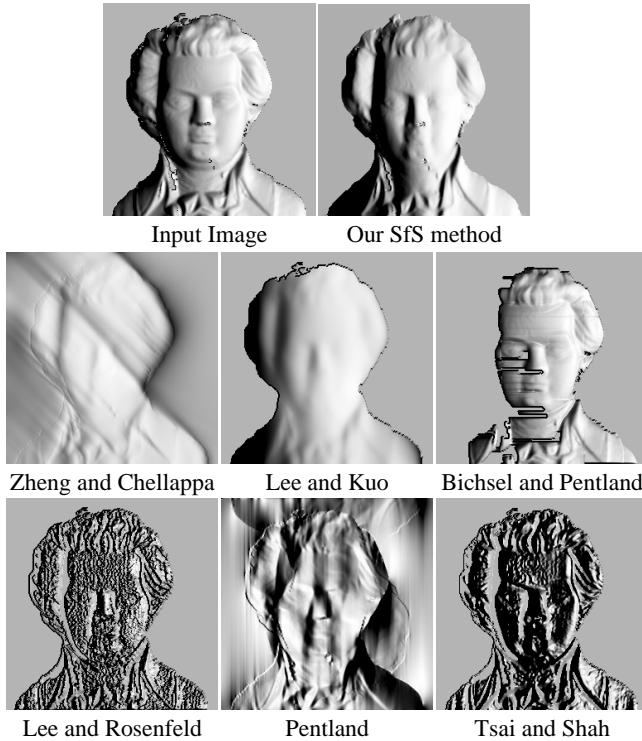
point light source was considered. While good results can be produced, the location of the light source is assumed to be coincided with the center of projection, and the intrinsic parameters of the camera (focal length and pixel size) are required. Both restrictions limit the class of images that can be handled.

With only a single, uncalibrated image available in our system, in this paper we thus resort to human user *interaction* to help resolve the bas-relief ambiguity. Although interactive SfS [Zeng et al. 2005] was proposed, where the user specifies normal constraints to regularize the SfS problem, this method is limited to perfect data where the scene should be accurately described by the reflectance model. All surface results in [Zeng et al. 2005] follow closely the Lambertian assumption. It is unclear if their method can be extended to non-Lambertian scenes, where a non-linear reflectance model will complicate the problem setting.

To gain a better insight on user interaction in single-view modeling, we turn to Koenderink’s seminal paper *Pictorial Relief* [1998], which states that human’s monocular perception requires more information than geometry to interpret the scene. Using single images of a female torso sculpture, he performed a study on a large number of human subjects. It was found that the consciousness corresponding to the pictorial relief (3D surface perceived from single images) is more or less the same for different persons, even under variable scene illumination (shading). Specifically, the assignment of local normals (or *relative depths*, referred to as “the direction of local depth gradients” in [Koenderink 1998]) are very consistent among all the subjects, while that of absolute depth magnitudes is rather imprecise. Because local normals can capture well high-frequency component of the underlying visible surface, they were utilized by ShapePalettes [Wu et al. 2007] in their design of the 2D interface for creating shapes with sparse, high-frequency details.

The low-frequency component is, on the other hand, difficult to recover because it is distorted or even lost during the image formation process (see the detailed frequency analysis in [Frankot and Chellappa 1988]<sup>1</sup>). Fortunately, it is easier to specify, because it consti-

<sup>1</sup>For example, using the same directional light, the *same* photo of a Lambertian sphere can be produced by rendering the 3D sphere, or its normal map, which is flat.



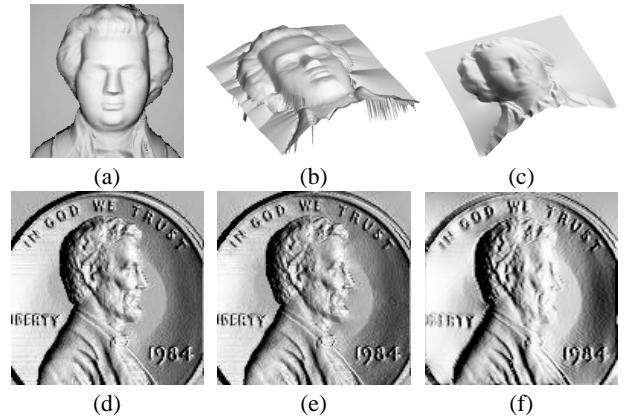
**Figure 3:** Comparison of SfS algorithms on initial normal estimation – classical method: The data set Mozart and the codes of the tested SfS algorithms are obtained from [Zhang et al. 1999]. The input image is rendered using the lighting direction  $\mathbf{L} = (1, 0, 1)^T$ . Other settings of the experiment are based on the discussion in [Zhang et al. 1999]. The output normal maps are rendered with the same lighting direction using Lambertian shading.

tutes the smooth global shape of the object. Noting from [Koenderink 1998] that the specification of absolute depth is inadvisable, we propose that the user specifies *relative normals* to edit and correct low-frequency errors. The use of relative information has shown successes in 3D mesh modeling: in [Sorkine et al. 2004], a set of useful 3D mesh editing operations were proposed which operates on the Laplacian of the mesh. In [Yu et al. 2004], an existing mesh can be edited by operating in the gradient domain.

### 3 Our Approach

In this section, we present a practical two-step approach to create high-quality surface normals using only small amount of user interaction. The two steps correspond to our two complementary contributions which, when taken together, produce some of the best normal reconstruction results to date from a single image of complex objects.

In the first step, our novel interactive SfS algorithm recovers high-frequency details from a single image. Our algorithm is largely automatic and produces good initial normal estimation for objects with nearly homogeneous albedos. In particular, this SfS algorithm globally distributes normal reconstruction errors over all the possible lighting directions, thus ameliorating the bas-relief ambiguity which biases the result toward the input. This strategy, however, makes the reconstruction error more pronounced in the low-frequency component, as shown in Figure 2(b). To correct these errors we propose a convenient editing tool which requires only simple and sparse markup to produce a transformation map (Fig-



**Figure 4:** Comparison of SfS algorithms on initial normal estimation – recent methods. (a) Input image Mozart rendered at the lighting direction  $\mathbf{L} = (0, 0, 1)^T$ . (b) and (c) show the surfaces respectively reconstructed by [Prados and Faugeras 2005] and our SfS method. (d) Input image Penny rendered at the lighting direction is  $\mathbf{L} = (1, 0, 1)^T$ . (e) and (f) show the normal maps respectively reconstructed by [Potetz 2007] and our SfS method.

ure 2(c)). Further, high-frequency surface details can be enhanced by using the same tool. This tool, which we call *rotation palette*, is a direct extension of shape palettes [Wu et al. 2007], where the sphere image is used for picking relative normals via 2D markups. Figure 2(d) shows our result after only a few sparse normal editing operations. Figure 2(e) is the surface reconstructed by using the normal map shown in Figure 2(d).

#### 3.1 Initial Normal Estimation

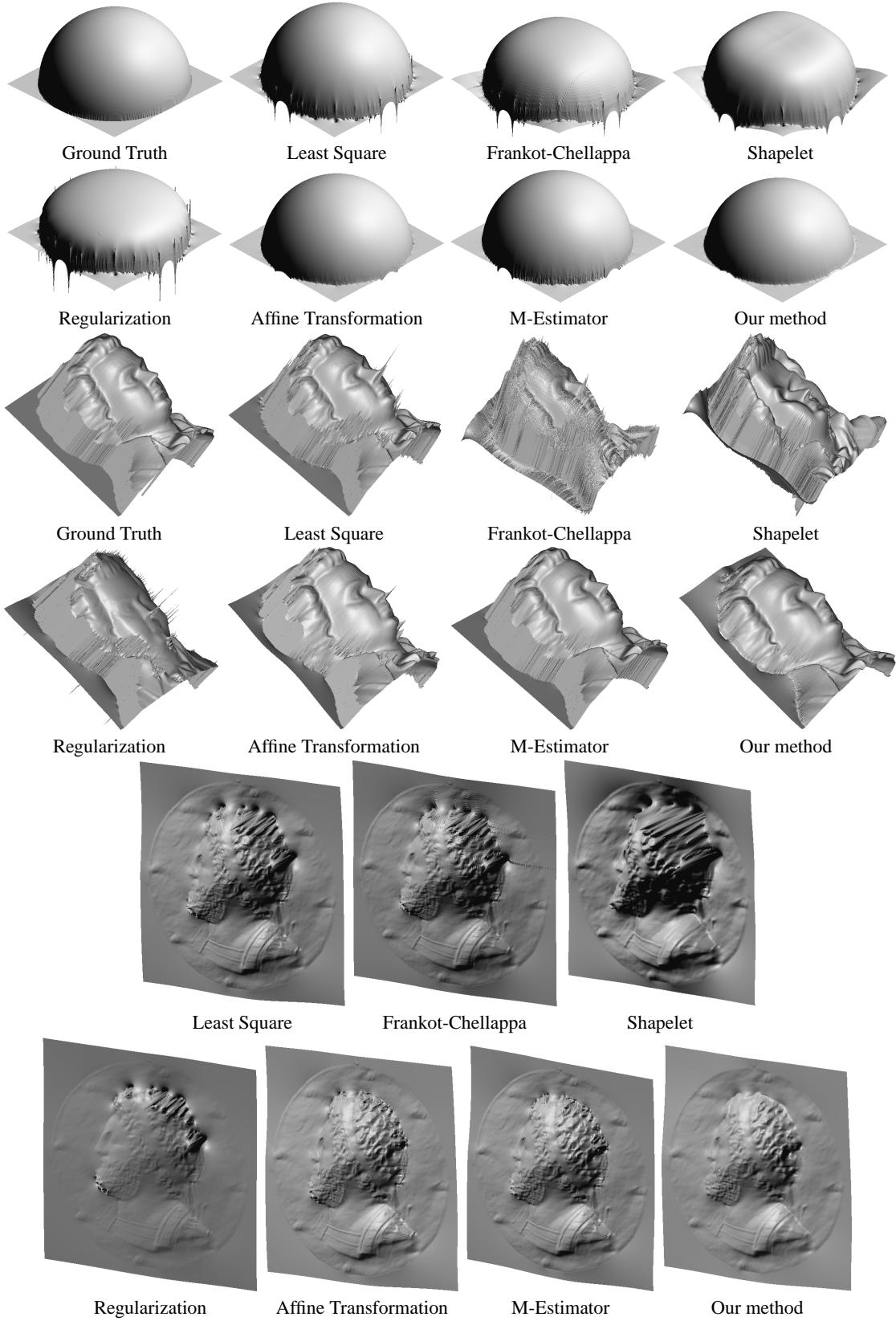
We first present our SfS algorithm for initial normal estimation. The better the initial normal map, the easier and more effective it will make the normal editing step that followed (section 3.2). We first studied some classical automatic SfS algorithms [Zhang et al. 1999] for which implementations are publicly available. The results are shown in Figure 3, indicating that the normal maps produced by these methods are not suitable for normal editing even for a Lambertian-shaded image. Comparison with more recent methods [Prados and Faugeras 2005; Potetz 2007] in Figure 4 indicates that although the results produced by these methods are more appealing, the restrictions on the light source, the restrictive requirement of intrinsic camera parameters [Prados and Faugeras 2005], and the long running time (typically one hour in [Potetz 2007]) make these methods unsuitable in our single-image interactive system.

Although our SfS method employs the Lambertian model, our experimental results on non-Lambertian objects are very good after our system has accepted a small amount of user input. The Lambertian imaging model is:

$$I = \rho \mathbf{N}^T \mathbf{L} \quad (1)$$

where  $I$  is the input image,  $\rho$  is the surface albedo,  $\mathbf{N} = (n_x, n_y, n_z)^T$  is a unit vector representing the surface normal, and  $\mathbf{L} = (l_x, l_y, l_z)^T$  is a unit vector representing the direction of a distant light source. All the terms on the right hand side of Eqn. 1 are unknown. To obtain the lighting direction  $\mathbf{L}$ , we require the user to assign normals to a few pixels ( $\geq 3$ ) and minimize the following energy:

$$E_1 = \sum_{i \in \mathcal{C}} \|I_i - \mathbf{N}_i^T \mathbf{L}'\|^2, \quad (2)$$



**Figure 5:** Comparison on surface-from-normals algorithms on height generation: The surfaces reconstructed by using various methods (the source codes of the tested algorithms are obtained from [Agrawal et al. 2006]). Our method attenuates the harmful effect of outliers by minimizing the surface curvature with osculating arc, and assumes the surface to be singly connected. In contrast, affine transformation and M-Estimator consider discontinuities in reconstruction. However, among all the tested methods, only our method produces the least distortion while achieving interactive speed in height generation.

	$(1, 1, 1)^T$	$(-1, 1, 1)^T$	$(-1, -1, 1)^T$	$(1, -1, 1)^T$
Lighting direction				
Ground truth image set $\mathbf{G}$				
Rendered image set $\mathbf{S}_1$				
Rendered image set $\mathbf{S}_2$				
$ \mathbf{G} - \mathbf{S}_1 $				
Per-pixel intensity residual	0.0024	0.3002	0.1740	0.3001
$ \mathbf{G} - \mathbf{S}_2 $				
Per-pixel intensity residual	0.0607	0.0666	0.0643	0.0659

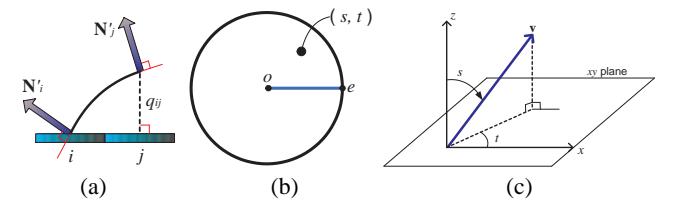
**Table 1:** Sphere – A study on the error distribution of our proposed SfS algorithm. The ground truth image, illuminated at the lighting direction  $(1, 1, 1)^T$ , is used as the input image to our algorithm. The normal map corresponding to image set  $\mathbf{S}_1$  is obtained by solving Eqn. 3. The normal map corresponding to image set  $\mathbf{S}_2$  is obtained after surface reconstruction by solving Eqn. 4 (that is, the complete SfS algorithm). It can be observed that  $\mathbf{S}_1$  looks faithful to the input when the output normal map is rendered using the same light as the input. But the error of the rendered image is quite large when the output is rendered at other lighting directions. After our surface reconstruction process, which is done by solving a Poisson equation derived from Eqn. 4, the biases in error are ameliorated: the errors are more evenly distributed among all lighting directions, as shown in the rendered set  $\mathbf{S}_2$ .

where  $\mathcal{L}$  is a set of user selected pixels and  $\mathbf{L}' = \rho\mathbf{L}$ . Then, the unit lighting direction is obtained by  $\mathbf{L} = \frac{\mathbf{L}'}{\|\mathbf{L}'\|}$ . This estimation only needs to be performed once. Using the estimated  $\mathbf{L}$ , we can compute the normals  $\mathbf{N}$  and the albedo  $\rho$  by minimizing the following energy:

$$E_2 = \sum_{i \in \mathcal{P}} \|\rho' I_i - \mathbf{N}_i^T \mathbf{L}\|^2 + \lambda \sum_{\{i, j\}} \|\mathbf{N}_i - \mathbf{N}_j\|^2 \quad (3)$$

where  $\mathcal{P}$  is the user selected region to be processed,  $\{i, j\}$  is a first-order neighbor pair,  $\lambda$  is a regularization factor, and  $\rho' = \rho^{-1}$ . The first term in the energy function measures the fitness of the imaging model and the second term enforces a smoothness constraint on the normals. Since the energy in Eqn. 3 is a quadratic function, it can be minimized using Gauss-Seidel method with successive over-relaxation. In each iteration, the unit-length constraint of  $\mathbf{N}_i$  is enforced by re-projecting the updated  $\mathbf{N}_i$  onto a unit sphere and  $\rho'$  is restricted in the range  $\rho' \geq 1$  since  $0 \leq \rho \leq 1$ .

We studied the results generated by minimizing Eqn. 3. Table 1 shows that the resulting normal map has a general bias toward the input lighting direction. Rather than introducing complex problem models which make the estimation non-linear, we propose to ame-



**Figure 6:** (a) Relative height estimation. The black arc is the fitted osculating arc, (b) the information encoded in the 2D view of a sphere, (c) an illustration of rotation.

liorate the effect of such bias by evenly distributing the errors across all directions, which can be achieved by solving a Poisson equation.

To evenly distribute the error, we reconstruct the height field  $\mathbf{H} = \{h_i\}$  of the object by minimizing the following energy (which turns out to be a Poisson equation), in which the lighting direction is absent:

$$E_3 = \sum_{\{i, j\}} ((h_i - h_j) - q_{ij})^2, \quad (4)$$

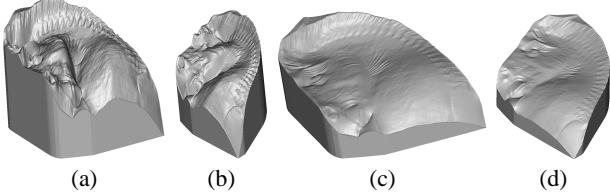
where  $h_i$  and  $h_j$  are the heights at  $i$  and  $j$ , and  $q_{ij}$  is the relative height between  $i$  and  $j$  on the surface. In previous work,  $q_{ij}$  is calculated from the surface gradient ( $\frac{\partial f}{\partial x} = \frac{n_x}{n_z}$ ,  $\frac{\partial f}{\partial y} = \frac{n_y}{n_z}$ ). However, the gradient will be undefined when the normal is perpendicular to the viewing direction (e.g., the object’s occlusion boundary in 3D is projected as the object’s silhouette on a 2D image). To avoid the gradient approaching to infinity (i.e.  $\frac{\partial f}{\partial x}$  or  $\frac{\partial f}{\partial y} \rightarrow \infty$ ), we calculate the *relative height*  $q_{ij}$  directly from normals by postulating a smooth connection. This strategy allows a better surface to be produced with less distortion, when the change of surface gradient is large, especially along object boundaries and crease curves. Figure 5 compares the results using our method and other surface-from-gradients algorithms on the same data sets. We detail the method in the following:

**The osculating arc constraint** Figure 6(a) illustrates how we calculate the relative height  $q_{ij}$  between two neighboring pixels  $i = (x, y)$  and  $j = (x + 1, y)$  along the  $x$  direction in the image. We first project the normals  $\mathbf{N}_i$  and  $\mathbf{N}_j$  onto a vertical plane, which is parallel to the  $x$  direction, to obtain two vectors  $\mathbf{N}'_i$  and  $\mathbf{N}'_j$ . Then, we fit an osculating arc [Gray 1997] between the two projected normals. This arc can be uniquely defined by  $\mathbf{N}'_i$  and  $\mathbf{N}'_j$  where their tangents touch the arc, resulting in a minimal curvature connection between  $i$  and  $j$ . To calculate  $q_{ij}$  for  $y$  direction, the same method is applied where the normal  $\mathbf{N}_i$  and  $\mathbf{N}_j$  are projected onto a plane parallel to the  $y$  direction. The use of such geometric smoothness as described allows us to avoid numerical instability due to ill-defined surface gradients  $\frac{\partial f}{\partial x}$  or  $\frac{\partial f}{\partial y}$ , which are typical of a complex surface containing orientation discontinuities. After surface reconstruction, we recompute the normals directly from the height field.

Using the surface reconstruction method described above, we can improve the initial estimated normals. However, the normal map is often problematic especially in the low-frequency parts. Next, we propose an approach on normal editing that allows the user to easily correct these errors. Moreover, the same tool can also be employed to further enhance high-frequency surface details when necessary.

### 3.2 RotationPalette: Interactive Normal Manipulation

Previous approaches allow a user to specify a desired normal map,  $\mathbf{N}'$ , from scratch (e.g. [Zhang et al. 2001; Wu et al. 2007]). However these approaches are not appropriate when an existing  $\mathbf{N}$  is available for incorporating low-frequency components. One may sug-



**Figure 7:** (a)–(b) show two views of a surface generated by globally scaling up the height values produced by our SfS result. (c) – (d) show two views of a surface generated by our SfS algorithm, after Gamma correction is applied to the input image. Both plausible alternatives fail to incorporate the satisfactory low-frequency component.

gest to repair the estimated  $\mathbf{N}$  by global adjustment, such as height scaling, or by Gamma correction on the input image. Unfortunately, these straightforward alternatives will fail, as demonstrated in Figure 7, on typical images exhibiting intricate surface details.

We propose to manipulate normals by specifying the *relative normal*  $\Delta\mathbf{N}$ , which is ideally equal to  $\mathbf{N}' - \mathbf{N}$ , as opposed to absolute normal. It is easier for a user to specify constraints in a smooth, low-frequency, relative normal map, and sparse markups can be propagated to the whole normal map easily by an optimization method.

Specifically, in our system, the user manipulates the normal  $\mathbf{N}$  by a *rotational transformation* to produce the rotated normal  $\mathbf{N}' = \mathbf{R}(\mathbf{N}; s, t)$ , where  $s$  and  $t$  are the slant and tilt for the rotation. This allows easy user interaction on a 2D interface, because the user can simply draw points or strokes inside a circle to provide sample  $(s, t)$ . Figure 6(b) shows a circle which corresponds the orthographic projection of a sphere and Figure 6(c) illustrates the relationship of slant  $s$  and tilt  $t$  in rotation, where  $s$  and  $t$  define the angle and direction of rotation respectively. With this, the vector at  $o$  produces zero rotation. Now, suppose we move from  $o$  to  $e$  along the blue line as shown, which corresponds to moving along the pertinent arc on the sphere. As the length  $|oe|$  increases, the angle of rotation also increases while the axis of rotation remains unchanged. Thus this 2D circle provides a convenient means for the user to control the strength of the transformation, which is measured by  $|oe|$ .

In actual implementation, we select a  $(s, t)$  from the sphere, and then convert it into a rotation matrix  $\mathbf{R}$ . We rotate  $\mathbf{N}$  by  $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{v})$  where  $\mathbf{r}_1 = -\frac{\mathbf{v} \times \mathbf{r}_2}{\|\mathbf{v} \times \mathbf{r}_2\|}$ ,  $\mathbf{r}_2 = \frac{\mathbf{v} \times \mathbf{a}}{\|\mathbf{v} \times \mathbf{a}\|}$ ,  $\mathbf{a} = (0, 0, 1)^T$ , and  $\mathbf{v} = (\cos t \sin s, \sin t \sin s, \cos s)^T$ . Note that the user only needs to specify a sparse set of rotational transformation in terms of  $(s, t)$ . We provide an optimization method to propagate the user inputs, by minimizing the following energy function w.r.t.  $\mathbf{v}_i$ , and produce a dense map  $\mathbf{V}$ :

$$E_3 = \sum_{i \in \mathcal{U}} \|\mathbf{v}_i - \mathbf{v}'_i\|^2 + \beta \sum_{\{i, j\}} \|\mathbf{v}_i - \mathbf{v}_j\|^2, \quad (5)$$

where  $\mathbf{v}'_i$  is the user selected rotation vector on the sphere image,  $\{i, j\}$  is a first-order neighbor pair,  $\mathcal{U}$  is the set of user specified pixels, and  $\beta$  is a regularization factor (the value is fixed to 0.005). Figure 2(c) visualizes one instance of the dense map  $\mathbf{V}$  rendered by using Lambertian shading. This is very similar to a typical normal map.

With the rotation method mentioned above, we provide two types of user interaction for editing the initial normal map:

**Surface control.** The user picks a point or draws a stroke to specify the desired rotations from a sphere. The rotation effect is

propagated by Eqn. 5. Figure 8 shows a simple example where the cylinder is too flat. If we want to pop up the cylinder, we just need to draw a pair of corresponding curves on the sphere to increase the surface curvature.

**Detail embedding.** User can enhance surface details by selecting a region of interest from the input image. Inside the selected region, the image gradients will be converted into a set of  $\mathbf{v}$ 's

$$\mathbf{v} = \frac{1}{\sqrt{1 + \left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}} \left( \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, 1 \right)^T \quad (6)$$

where the corresponding normals are rotated by these  $\mathbf{v}$ 's. We update the normals by  $\tilde{\mathbf{N}} = (1 - \alpha)\mathbf{N} + \alpha\mathbf{N}'$ , where  $\mathbf{N}'$  is the rotated  $\mathbf{N}$  and  $\alpha$  (default value is 0.2) controls the contribution of the new normal. An example is shown in Figure 11 (third row, example *Relief*) where the surface details of the whole portrait was enhanced. This can also be used to synthesize surface details when unwanted structures are removed. A demonstration will be shown in Section 3.3.

**Interface: Shape Palettes vs Rotation Palette.** Our user interface, aptly called *rotation palette*, is clearly inspired by shape palettes [Wu et al. 2007], although there are fundamental differences in *what* to specify and transfer using the sphere palette. Table 2 compares rotation palette with shape palettes.

	Shape palettes	Rotation palette
palette	any	sphere
what to specify	absolute normal $\mathbf{N}'$	relative normal $\Delta\mathbf{N}$
how to specify	2D strokes	2D strokes

**Table 2:** Shape palettes [Wu et al. 2007] and rotation palette [this paper].

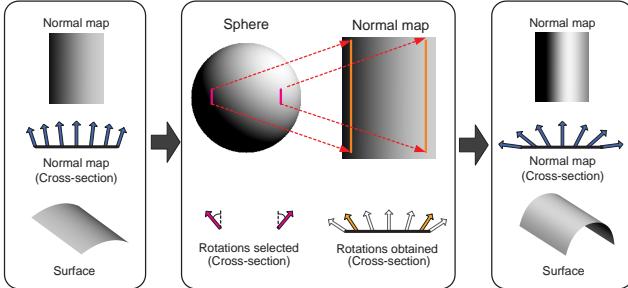
The user uses ShapePalettes to interactively transfer *absolute* normals  $\mathbf{N}'$  from a sphere palette to model the target shape. Therefore, to interactively model Venus's wrinkled garment, the user needs to mark several ridges and valleys on the sphere palette. However, it is still impossible to model subtle details spread all over the surface.

In RotationPalette, we use a sphere image as palette to mark up *relative* normals to refine a normal map. In the same vein as successful mesh editing systems [Sorkine et al. 2004; Yu et al. 2004], which operate in the relative quantities, interacting with a sphere palette by specifying relative normals is the key to success of our interactive system, allowing drastic reduction in the amount of user markup on the sphere. (See Table 3 and the supplemental video.)

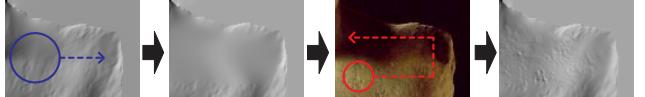
### 3.3 A Real Example

Figure 9 shows the example *Venus*. It may be difficult to notice the problem at first glance because the rendered normal map looks very similar to the original image. The normal map is rendered with lighting direction  $(-1, 1, 1)^T$ . When we examine the normal map more closely by changing the lighting direction (see supplemental video), we can notice that the three zoom-in normal regions exhibit obvious problems. The left shoulder is shadowed, the chest is too flat, and the whole left-hand-side of the sculpture is too bright. Because of these problems, the shadow will be mistaken as part of the structure and the reconstructed surface will be very flat. More minor problems are also revealed when the lighting direction is changed. For clarity of illustration, we use these three example regions to describe the necessary interactions as the following.

**Surface control by strokes.** To pop up a flat surface region, we want to increase the magnitude of the surface's slope. This means that we should select a rotation that increases the slant of the normal



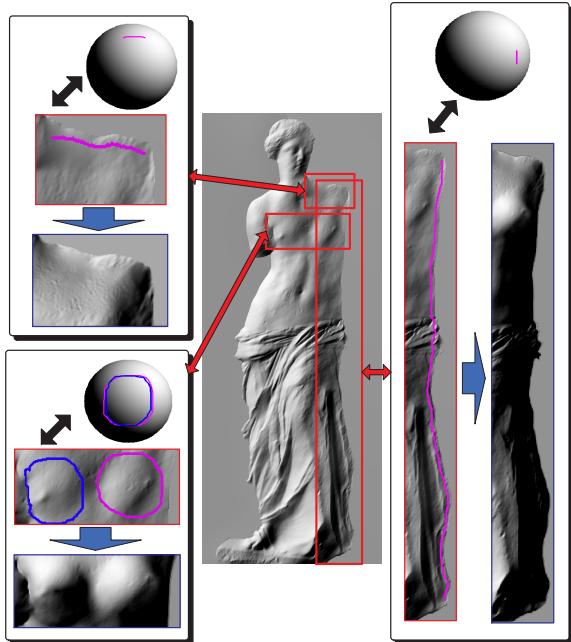
**Figure 8:** An example on popping up a flat surface. The vectors selected by the pink curves will be copied to the orange curves, which will be propagated to the whole image (white vectors) by Eqn. 5 where the normals rotated at the corresponding pixels to produce a final normal map, as shown in the right column.



**Figure 10:** Blurry brush (blue) and gradient brush (red).

		total processing time (sec)	no. of strokes/brushes	total interaction time (min)
Horse	533 × 476	~ 2	11	~ 3
Venus	422 × 1060	~ 2	6	~ 1
Man	420 × 560	~ 2	19	~ 10
Lady	500 × 714	~ 2	9	~ 5
Relief	549 × 550	~ 2	15	~ 10
Bird	533 × 606	~ 2	16	~ 10

**Table 3:** Summary of processing and interaction time. Refer to the supplemental video for the actual interaction of some examples. The processing time for our automatic SfS is measured on a laptop computer with Pentium 4 3GHz CPU and 512 MB memory. The interaction time required depends on the complexity of the object. The processing time after each markup is almost instant (< 1 sec).



**Figure 9:** Enhancing the normal map of Venus: three examples on stroke markups on rotation palettes.

in the region of interest. We show the corresponding actual markup from three zoom-in views in Figure 9.

**Details embedding by brushes.** To remove unwanted structures caused by shadows or highlights, the user employs the blurry brush, which is implemented by applying Gaussian kernel to remove problematic regions (see Figure 10). To add back details removed by applying the blurry brush, the user uses the gradient brush, which converts image gradient into  $\mathbf{v}$  by Eqn. 6 (and finally  $\mathbf{R}$ ) to embed the corresponding image detail. The user can change lighting direction to visualize the normal map, and apply one of these brushes until s/he is satisfied with the refinement result.

## 4 Results

Table 3 tabulates the processing and user interaction times. Our single image approach does not handle occlusion, so the underlying model is assumed to be a singly-connected surface. Therefore, surface distortion will be apparent for angular or non-orthographic views. For the following results, we show the estimated normal map and the surface model. The surface model was pasted on a

base whose height is equal to the maximum height of the surface. This will indicate roughly the viewpoint of the original input when a novel view of the surface is displayed.

**Horse.** Figure 1 shows an example of a horse. Note the fine details preserved in our reconstruction: the braided mane, the eyebrows, the wrinkles on the neck, the nostril are all faithfully recovered.

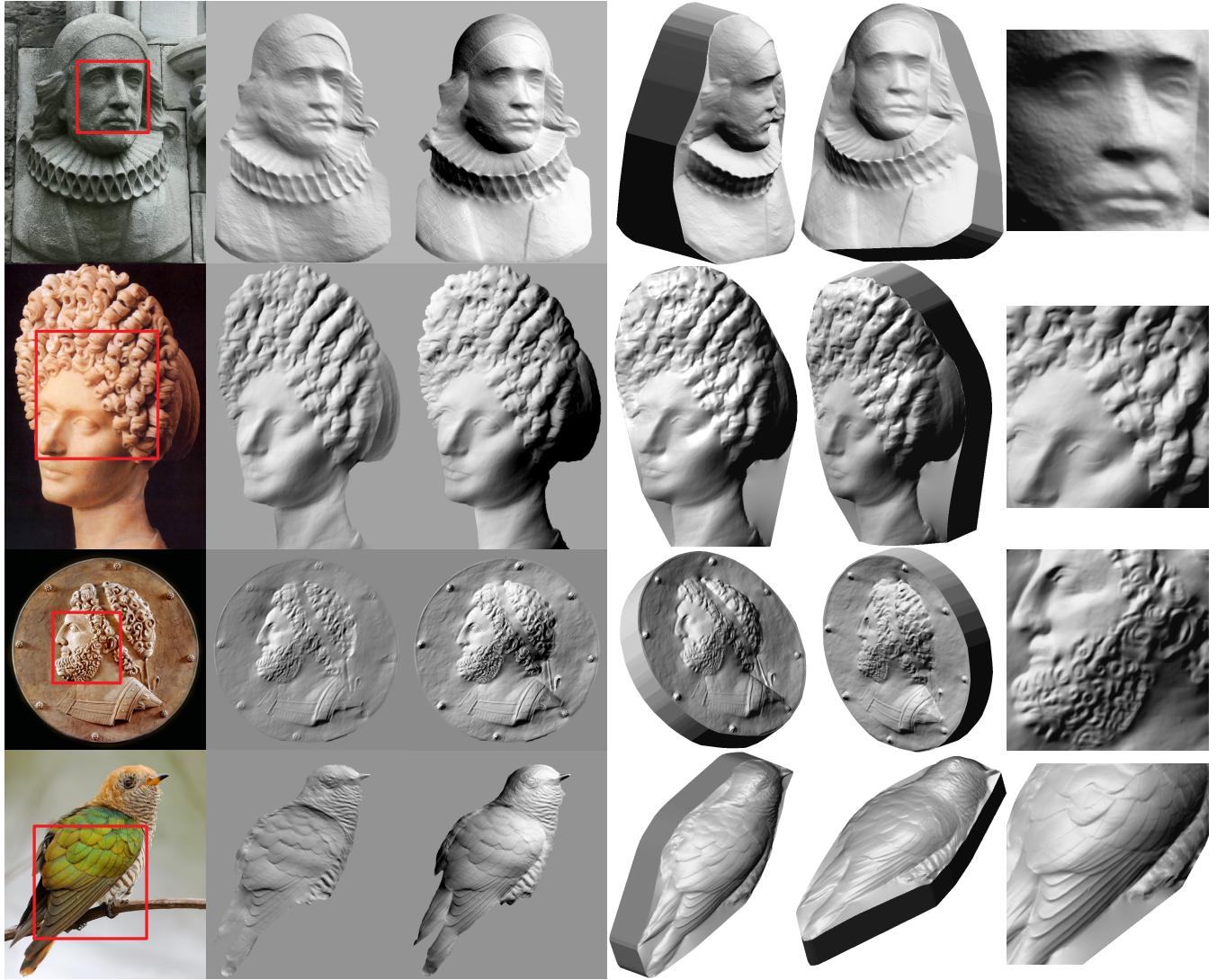
**Man.** Figure 11 (first row) shows the normal map and two views of the recovered surface of the original high relief, which was carved from a very rough material. Although the shading is very weak and unclear, our system can still pop up the entire surface. Note in particular the profile view of the recovered surface, where the nose and the chin look undistorted. The concavities of the pleated collar are not recovered well due to the shadows under the collar.

**Lady.** Figure 11 (second row) shows the results on a 3D sculpture of a lady, which is illuminated by a single light. The complex hairs pose a challenge. Note the faithfulness of our reconstruction. The surface suffers from some distortion due to the off-axis view of the sculpture in the input photograph.

**Relief.** Figure 11 (third row) shows the results for a bas-relief, which was illuminated by two lights when the image was captured: one on the left while the other on the right. This two-light configuration results in an erroneous light source estimation by our automatic SfS algorithm. Despite that, our results are still very good. Note the complex curly hairs, which are not easy to model in existing automatic or single image interactive modeling systems.

**Bird.** Figure 11 (fourth row) shows the results for a bird. Although the surface material exhibits non-homogeneity, the color contrast of the bird is small. Our method produces a decent normal map/surface where the fine details are preserved. However, our method cannot handle the complex feather structures of the bird, where the reconstructed surface looks a bit problematic.

**Nutcracker.** Figure 12 shows a result of a container to demonstrate the limitation of our method, where the color contrast of the object is large. In this case, our surface produced suffers severe distortion due to the single albedo assumption: high contrast area will be considered as sharp change in surface orientation.



**Figure 11:** From left to right: input image, initially estimated normal map, final normal map after editing, two views of the surface, and one zoom-in view of the surface at the input view with the corresponding region marked on the input image.

## 5 Discussion and Conclusion

We propose an interactive approach for reconstructing surface normals of an object in a single image. Our approach starts with a SfS algorithm that produces good initial normals for a large class of objects. Our SfS algorithm globally distributes errors to reduce biases toward the input light, while leaving most noticeable errors mainly in the low-frequency part. These errors can be easily corrected by the user. We use rotation palette, a very convenient 2D user-interface, to edit and correct a normal map, where the user merely marks up rotational transformation on a normal image of a sphere. This allows easy and intuitive manipulation using relative normals.

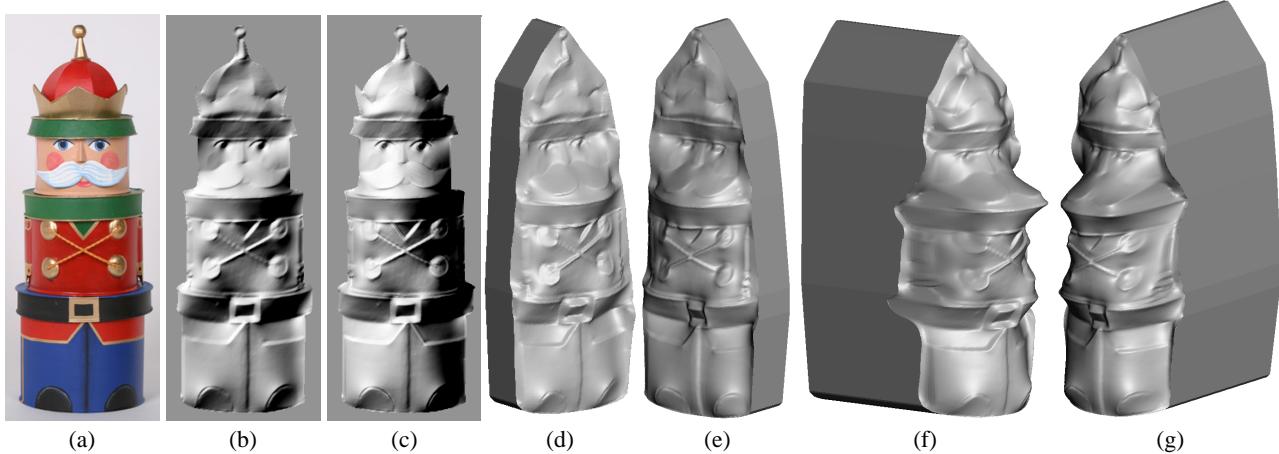
Directional lighting is assumed in our automatic SfS algorithm. In practice, our SfS algorithm is insensitive to the choice of pixels used to estimate the directional light. In our experiments, all sample pixels for light estimation were manually selected. Similar to other single-view and image-based approaches, occlusion and oblique views limit the reconstruction of invisible surface.

Presently, our system is designed for reconstructing normals from

single images of objects which are made of homogeneous material with relatively little textures and albedos. As demonstrated in our results and supplemental video, while the user’s markup need not be unique nor very accurate, which is a feature inherited from Shape Palettes [Wu et al. 2007], very good results can still be obtained. It may be possible for an automatic approach that roughly labels regions requiring normal refinement, while leaving to the human user the decision on the strength of transformation (i.e., how much a normal should be rotated via marking up on a sphere image), thus further reducing the amount of user markup.

## Acknowledgments

We would like to thank the anonymous reviewers for their discussion and constructive comments. Many thanks to Michael S. Brown for his fruitful suggestions and the help to improve the manuscript. Chi-Keung Tang acknowledges the support by the Research Grant Council of Hong Kong Special Administration Region, China: 620008.



**Figure 12:** Nutcracker – High color contrast (for illustrating the limitation of our method): (a) Input image. (b) Normal map generated by the SfS method only. (c) Normal map generated by our full method. (d)–(e) Two views of the reconstructed surface from normal map (b). (f)–(g) Two views of the reconstructed surface from normal map (c). In this example, our surface suffers severe distortion due to the single albedo assumption: high-contrast area will be considered as sharp change in surface orientation.

## References

- AGRAWAL, A., RASKAR, R., AND CHELLAPPA, R. 2006. What is the range of surface reconstructions from a gradient field? In *ECCV06*. <http://www.umiacs.umd.edu/~aagrawal/eccv06/RangeofSurfaceReconstructions.html>.
- BELHUMEUR, P. N., KRIEGLMAN, D. J., AND YUIILLE, A. L. 1999. The bas-relief ambiguity. *Int. J. Comput. Vision* 35, 1, 33–44.
- CRIMINISI, A., REID, I., AND ZISSERMAN, A. 2000. Single view metrology. *IJCV* 40, 2 (November), 123–148.
- FANG, H., AND HART, J. C. 2004. Textureshop: texture synthesis as a photograph editing tool. In *SIGGRAPH '04*, 354–359.
- FRANKOT, R., AND CHELLAPPA, R. 1988. A method for enforcing integrability in shape from shading algorithms. *PAMI* 10, 4 (July), 439–451.
- GINGOLD, Y., AND ZORIN, D. 2008. Shading-based surface editing. *ACM Transactions on Graphics (TOG)* 27, 3.
- GRAY, A. 1997. Osculating circles to plane curves. In *Modern Differential Geometry of Curves and Surfaces with Mathematica, 2nd ed.*, Boca Raton, FL: CRC Press, 111–115.
- HOIEM, D., EFROS, A. A., AND HEBERT, M. 2005. Automatic photo pop-up. *ACM Trans. Graph.* 24, 3, 577–584.
- HORN, B., AND BROOKS, M. 1989. *Shape from Shading*. The MIT Press.
- HORRY, Y., ANJKYO, K.-I., AND ARAI, K. 1997. Tour into the picture: using a spidery mesh interface to make animation from a single image. In *SIGGRAPH '97*, 225–232.
- KARPENKO, O. A., AND HUGHES, J. F. 2006. Smoothsketch: 3d free-form shapes from complex sketches. *ACM Trans. Graph.* 25, 3.
- KHAN, E. A., REINHARD, E., FLEMING, R. W., AND BÜLTHOFF, H. H. 2006. Image-based material editing. In *SIGGRAPH '06*, 654–663.
- KOENDERINK, J. 1998. Pictorial relief. *Royal* 356, 1740, 1071–1086.
- KOVESI, P. 2005. Shapelets correlated with surface normals produce surfaces. In *ICCV05*, 994–1001.
- POTETZ, B. 2007. Efficient belief propagation for vision using linear constraint nodes. In *CVPR 2007: Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, Minneapolis, MN, USA.
- PRADOS, E., AND FAUGERAS, O. 2005. Shape from shading: A well-posed problem? In *CVPR05*, II: 870–877.
- PRASAD, M., ZISSERMAN, A., AND FITZGIBBON, A. 2006. Single view reconstruction of curved surfaces. In *CVPR06*, II: 1345–1354.
- SORKINE, O., LIPMAN, Y., COHEN-OR, D., ALEXA, M., RÖSSL, C., AND SEIDEL, H.-P. 2004. Laplacian surface editing. In *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, Eurographics Association, 179–188.
- WU, T.-P., TANG, C.-K., BROWN, M. S., AND SHUM, H.-Y. 2007. Shapepalettes: interactive normal transfer via sketching. In *SIGGRAPH '07*, 44.
- YU, Y., ZHOU, K., XU, D., SHI, X., BAO, H., GUO, B., AND SHUM, H.-Y. 2004. Mesh editing with poisson-based gradient field manipulation. *ACM Trans. Graph.* 23, 3, 644–651.
- ZENG, G., MATSUSHITA, Y., QUAN, L., AND SHUM, H. 2005. Interactive shape from shading. In *CVPR05*, I: 343–350.
- 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. <http://server.cs.ucf.edu/~vision/faculty/shah.html>.
- ZHANG, L., DUGAS-PHOCION, G., SAMSON, J., AND SEITZ, S. 2001. Single view modeling of free-form scenes. In *CVPR01*, I:990–997.