Bas-Relief Modeling from Normal Images with Intuitive Styles

Zhongping Ji, Weiyin Ma, *Member, IEEE,* Xianfang Sun

Abstract—Traditional 3D model-based bas-relief modeling methods are often limited to model-dependent and monotonic relief styles. This paper presents a novel method for digital bas-relief modeling with intuitive style control. Given a composite normal image, the problem discussed in this paper involves generating a discontinuity-free depth field with high compression of depth data while preserving or even enhancing fine details. In our framework, several layers of normal images are composed into a single normal image. The original normal image on each layer is usually generated from 3D models or through other techniques as described in this paper. The bas-relief style is controlled by choosing a parameter and setting a targeted height for them. Bas-relief modeling and stylization are achieved simultaneously by solving a sparse linear system. Different from previous work, our method can be used to freely design bas-reliefs in normal image space instead of in object space, which makes it possible to use any popular image editing tools for bas-relief modeling. Experiments with a wide range of 3D models and scenes show that our method can effectively generate digital bas-reliefs.

Index Terms—Bas-relief, Normal image, Relief style, Feature preserving, Layer-based.

1 Introduction

Relief is a form of sculpture in which a solid piece of material is carved so that figures emerge from a background, as though they are trapped in the wood, stone, metal or other materials. There are two main types of reliefs: high relief and bas-relief (also called low relief). High relief is a type of sculpture where the figure stands out further from the ground with the most prominent elements of the composition being undercut, while bas-relief has a shallower overall depth in comparison with high relief. This paper focuses on bas-relief.

Bas-reliefs are commonly seen throughout the world, for example, on the walls of historical sites or monumental buildings. This sculpture technique has been practiced for thousands of years, and has been used independently in many cultures including ancient Egypt, India, Greece, China, Persia, etc. A bas-relief may use any medium or technique of sculpture, but stone carving and metal casting are the traditional ones. Nowadays, bas-reliefs are commonly used in architecture, industrial design and handiwork. However, even with the development of computer-aided-design, the design of bas-reliefs remains mainly in the hands of artists. Recently, the problem of automatic generation of bas-reliefs from 3D scenes has received great attention. The idea is straightforward: a flattened sculpture is produced on some base surface, for instance, portraiture on coinage. The overall range of depth of the elements in the sculpture is highly compressed. Parallel or perspective effects may also be used in bas-reliefs. A bas-relief usually has a single z depth for each x-y position, which can be treated as a height map. The remaining problem for automatic generation of bas-reliefs is how to compress the depth of a 3D scene onto a view plane.

In this paper, we present a novel method based on normal images. Our goal is to develop a simple but flexible method for bas-relief generation which clearly preserves or even enhances visible shape details. We present a tool to assist artists in designing bas-reliefs on computers. Furthermore, we classify bas-reliefs into two types from the point of view of appearance, the round bas-relief and the flat bas-relief. As shown in Fig. 1, the round bas-relief (see Fig. 1(b)) whose middle portion is always elevated the most, is somewhat plumper than the flat bas-relief (see Fig. 1(c)). The flat bas-relief possesses a narrow compressive depth range, which makes the prominent part nearly on a plane. In this paper, we consider the round bas-relief and the flat bas-relief as different design styles of bas-reliefs.

Contributions. We developed a simple and intuitive modeling technique for creating digital bas-reliefs. The predominant feature of our method is that it is able to produce different styles of bas-reliefs and permits the design of bas-reliefs in normal image space rather than object space. The specific contributions of this paper are:

- **Free from depth discontinuity**. Based on normal images, our method is intrinsically free from depth discontinuity, and it is not necessary to explicitly remove depth intervals at height discontinuities.
- **Bas-relief stylization**. Styles of bas-reliefs receive little attention before [1], [2]. We formalize this problem as a quadratic optimization problem which controls the global shape and fine details. It gives a smooth transition

- *Zhongping Ji is with the School of Computer Science, Hangzhou Dianzi University, Hangzhou, China. Email: jzp@hdu.edu.cn.*
- *Weiyin Ma is with the Department of Mechanical and Biomedical Engineering, City University of Hong Kong, Hong Kong, China. Email: mewma@cityu.edu.hk.*
- *Xianfang Sun is with the School of Computer Science and Informatics, Cardiff University, Cardiff, UK. Email: Xianfang.Sun@cs.cardiff.ac.uk.*

Fig. 1: Creating different styles of bas-relief from a composite normal image using our method: (a) a composite normal image from a fish model and the squama of a dragon model; (b) a resulting round bas-relief from the new normal image; (c) a resulting flat bas-relief; and (d) illustration of (b) and (c) from a different viewpoint.

between different styles.

• **Normal image space and layer-based framework**. In our method, one can design bas-reliefs in normal image space instead of in object space, which makes it possible for us to use image editing tools for bas-reliefs editing. Due to the merit of normal images, we propose a layer-based bas-relief modeling framework, which benefits from image editing tools and possesses the reusability of existing bas-relief designs.

2 Related Work

In this section, we briefly summarize the state-of-the-art approaches. Digital bas-relief generation is a young research topic in the computer graphics field. Little literature studied the problem of bas-relief [3] generation based on 3D scene in the past decade. Cignoni et al. [4] treated bas-relief generation as a problem of compressing the depth of a 3D scene onto a view plane. The main idea of their method is that the compression ratio is different with the distance between the observer and the projected point. Their principal rule is treating the 3D scene as a height field from the point of view of the camera, which is followed by the subsequent literature. An advantage of this treatment is that we can easily borrow some methods developed for tone mapping of High Dynamic Range (HDR) images [5]. For bas-reliefs, depths take place of the intensities in HDR image. Weyrich et al. [6] proposed an HDR-based method for constructing digital bas-reliefs from 3D scenes. They did not compress the depths directly, but nonlinearly compress the gradient magnitude to remove depth discontinuities. The bas-relief was then reconstructed by integrating the gradient field in a least-squares sense. It also permits the control of different scale features. Combining with linear rescaling, unsharp masking of gradient magnitude, Kerber et al. [7] proposed a feature preserving bas-relief generation method based on the depth compression of range images. The new depth field is reconstructed from the rescaled derivatives of the initial range image. Using four parameters, one can steer the compression ratio and the amount of details to be perceivable in the outcome. Kerber et al. [8] also presented a filtering approach which preserves curvature extrema during the compression process. In this way it is possible to handle complex scenes with fine details. Song et al. [9] generated bas-reliefs on the discrete differential coordinate domain, combining the concepts of mesh saliency, and shape exaggeration. Bian and Hu [10] proposed a method based on gradient compression and Laplacian sharpening, which produces bas-reliefs with well-preserved details. Inspired by the relations among HDR, histogram equalization and bas-relief generation, Sun et al. [11] adapted an AHE (adaptive histogram equalization) method to depth compression, which provides a new method of bas-relief generation. This method produces high quality bas-relief and preserves surface features well.

All the above methods used 3D models as input to produce bas-reliefs. Recently, some bas-relief generation methods were proposed with 2D images as input. Alexa et al. [12] proposed an approach to automatically generate relief surfaces that reproduce desired images under directional illumination. Given several lighting directions and desired images, a relief surface is computed which will generate radiance distributions similar to the desired images on diffusely reflecting surfaces. Li et al. [13] presented a two-scale approach for bas-relief estimation from a single image, aiming at restoring brick and stone relief from their rubbing images in a visually plausible manner. Wu et al. [14] developed an approach of producing bas-reliefs from human face images. They first created a bas-relief image from a human face image, and then used shape-from-shading (SfS) approach on the bas-relief image to construct a corresponding bas-relief. They trained and used a neural network to map human face images to bas-relief images, and applied image relighting technique to generate relit human face images for bas-relief reconstruction.

Our method starts from a 3D object or scene, and operates on the normal image of the scene. Different from previous 3D model based methods [4], [6], [7], [8], [9], [10], [11], our method avoids dealing with the depth discontinuities explicitly by resorting to normals. It is well known that normals play an important and essential role in real-time rendering. Normal is also essential in the inverse problems of rendering, namely photometric stereo and shape from shading [15]. Photometric stereo is a technique in computer vision for estimating the surface normals of objects by observing the object under different lighting conditions. The special case where the data is a single image is known as shape from shading. Normals are also important in 3D modeling. Recently, a sketch based interactive normal transfer method for 3D modeling was proposed as Shape Palettes [16]. Our method is somewhat related to this work, but our goal is to create bas-reliefs via solving the fundamental problems which include preserving the appearance for orthogonal views and squashing the discontinuity gaps. It is essentially different from above techniques. Our method is motivated to construct a height field with similar appearance of existing surfaces (normals) ***under a height constraint and free from depth gaps***.

Different from the 2D image based methods [12], [13], [14], our method does not operate on natural images, but images of surface normals. Both our method and SfS algorithms have a step of constructing height fields from surface normals. However, traditional SfS algorithms are not suitable for bas-relief modeling because they tend to reconstruct fully extended objects whose proportions are correct in 3D space. Furthermore, the luminance, color and texture in a natural image usually do not reflect the geometric feature properties properly, which makes the algorithms based on natural images only work well for objects with simple materials. Although it is cheap to directly generate bas-reliefs from natural images, the quality of those bas-reliefs is yet not satisfactory. However, our algorithm focuses on preserving the appearance of a 3D scene for the orthogonal view and removing depth gaps under a height constraint. Our algorithm introduced different constraints from the general SfS algorithms to compress the height field. In addition, it does not need to calculate normals from images as SfS algorithms do. Taking 3D models as input, and equipped with image editing tools, our method can generate high-quality bas-reliefs that satisfy many applications. Admittedly, it is more expensive to get 3D models than 2D images. However, the cost of obtaining 3D models will not be a problem as 3D scanning techniques progress.

Problem statement and our approach. The major academic work to date has been devoted to achieve the necessary compression of depths. However, the results so far are not fully satisfactory and there is a lot of space for further improvement.

• The principal rule of previous methods is to treat a 3D object or scene as a height field from the point of view of the camera. An important step of most of the previous methods is to remove unnecessary depth discontinuities. They rely on one or more parameters to attenuate the discontinuities. We reconstruct the heights from normals, which provides a local description of the surface. For our method, the removal of unused depth intervals between back objects and front objects is unnecessary.

• The previous methods rarely consider different design styles of bas-reliefs. We follow the rule that treats the bas-relief as a height field, but our method permits convenient selection of different design styles.

• Because previous methods directly deal with a 3D scene, it is hard to edit the resulting bas-relief. Furthermore, the reusability of the existing design has not been considered before. In our framework, normals of a 3D scene are treated as a rendering image which can be reused or edited freely. Because the normal encodes the relative variation of surface, layers of normal images can be overlapped or blended to form a new normal image.

3 Bas-Relief Modeling and Stylization

3.1 Algorithm overview

Bas-relief presents the unique challenge of squeezing shapes into a nearly-flat surface while maintaining the fine details of the 3D scene as much as possible [6]. The fundamental problems in bas-relief modeling include preserving the appearance for orthogonal views and squashing the discontinuity gaps.

To solve the fundamental problems in bas-relief modeling, we use normal images and decompose bas-relief modeling into two terms:

1) restore the geometry from normals which encode the fine details and are intrinsically free from discontinuities; and
2) squash foreground objects down against background objects while preserving or enhancing fine details.

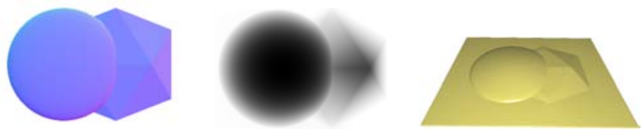
Moreover, artists may have a puzzle in mind: Given a 3D scene, how to produce different styles of bas-reliefs? As a preliminary solution to this problem, we present two basic styles: round bas-reliefs and flat bas-reliefs. Fig. 1 gives an example, showing that our method constructs new surfaces using some normal patches of different 3D models and produces two styles of bas-relief sculptures. Our solution of 3D bas-relief modeling and stylization tries to preserve fine details under a height constraint.

The automatic generation of bas-relief can be formulated as follows: Given a normal field $\boldsymbol{N}$, our goal is to create a height field H whose details are similar with that of $\boldsymbol{N}$ for the orthogonal view:

$$H(u,v) = \min_{H'} \iint (\mathcal{D}(\boldsymbol{N}, H') + \lambda \mathcal{F}(H', \theta)) du dv. \quad (1)$$

This energy function is a sum of two terms: one that measures the detail similarity of $\boldsymbol{N}$ and H' ($\mathcal{D}$ means 'Detail'), and the second one tries to fix a height constraint ($\mathcal{F}$ means 'Flatten'). θ is a threshold used to control the height of the resulting bas-relief. The relative contributions of two items are controlled by the parameter λ.

Our method begins with a normal image which is regularly sampled from a general polygonal 3D scene, created using a bump mapping technique, image tools or composed of different normal image layers. A height



(a) (b) (c)

Fig. 2: Generating digital bas-relief from a normal image: (a) starting from a 3D scene (a sphere and an icosahedron), we render a view-dependent normal image; (b) we then calculate the *Laplacians*, and integrate it to get a height field (black indicates large value); and (c) create a bas-relief by solving Equation (1).

field is obtained via a quadratic optimization. A simple and clear example is given in Fig. 2. A blue-purple-ish normal image is sampled from an input scene first (see Fig. 2(a)), and a height field (see Fig. 2(b)) is derived using our method.

3.2 *Laplacians* from normals

Given a 3D scene, we try to retain fine details for orthogonal views by resorting to the normals to avoid dealing with the depth discontinuities explicitly. We compute the heights by approximating the Laplacians using the least-squares method, so it can be considered as 2nd order continuity in this approximation sense.

***Laplacians* in a normal image**: Our motivation is to create a height field $H(u, v)$ whose normals resemble a given normal image $\boldsymbol{N}$ as much as possible. However, to preserve the normals directly will result in a non-linear optimization procedure. From another point of view, the local normal information can be used to represent the local detail on the surface. We encode the local details for orthogonal views as the *Laplacians* of the normal field.

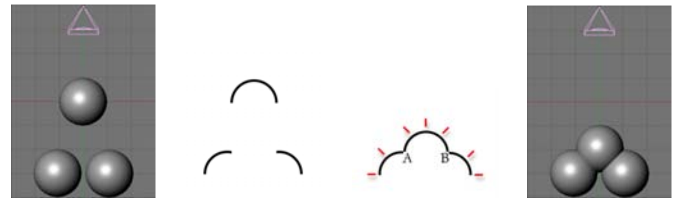
Given a normal image $\boldsymbol{N}$, we suppose its underlying surface is a height field H' and we define its *Laplacians* $\mathbb{L}(\boldsymbol{N})$ as follows,

$$\mathbb{L}(\boldsymbol{N}) = \mathcal{L}(H') = Div(\nabla H'), \tag{2}$$

where $\mathcal{L}(H')$ and $\nabla H'$ are the *Laplacians* and the gradient vector of H' respectively:

$$\nabla H' = (\frac{\partial H'}{\partial u}, \frac{\partial H'}{\partial v}) = (\frac{\boldsymbol{N}_x}{\boldsymbol{N}_z}, \frac{\boldsymbol{N}_y}{\boldsymbol{N}_z}). \tag{3}$$

Depth discontinuities in the 3D scene **must not give rise to depth discontinuities in the bas-relief**, otherwise, an extrusion from the foreground object to the background object would show artificial effect when viewed even slightly from the side. Our algorithm does not compute the gradient vector using separate triangles, which avoids introducing large gradient discontinuities especially on occluding boundaries where two neighbor pixels are sampled from two separate triangles. From this perspective, our method can be regarded as **intrinsically free from depth discontinuity**.



(a) (b) (c) (d)

Fig. 3: Depth discontinuity free intrinsically: (a) input scene; (b) depth discontinuities; (c) visible normals; and (d) an underlying scene with the same visible normals as (a).

(a) (b) (c)

Fig. 4: Bas-relief generated from a normal image, with normals preserved well. (a) A normal image of a 3D model; (b) resulting bas-relief from (a); and (c) normals of (b).

Fig. 3 shows a scene of three spheres apart from each other. While looking down from top, we just catch three patches of these surfaces. The depth is discontinuous (see Fig. 3(b)). With regard to this scene, previous work focuses on eliminating depth discontinuities (either manually or automatically).

However, we manage to remove these gaps from a different perspective. We seek for a *continuous surface* whose normals are formed by the counterpart of the visible surface patches. From the point of view of the observer, normals in Fig. 3(c) are from the underlying surface in Fig. 3(d). A more articulate example is shown in Fig. 4. Fig. 4(a) is a low-resolution (200×200 pixels) normal image sampled from a 3D model. The bas-relief (Fig. 4(c)) produced using our method displays similar appearance for orthogonal views with the input (Fig. 4(a)). However, if a normal image from a 3D scene has not been downsampled or smoothed,it may not exactly correspond to an underlying height field, especially around the occlusion boundaries with abrupt normal changes. Our optimization process finds a solution in the least squares sense. Consequently, deformation or extension in 3D space may arise to obey the continuity constraints applied by the integral. For example, a mild and acceptable deformation occurs near the occlusion boundary in Fig. 4(b), which makes the occluded part behind the self-intersection region 'visible' in the resulting bas-relief. We have not downsampled the normal image or smoothed the *Laplacians* for this example.

(a) (b) (c)

Fig. 5: Bas-relief examples using different energy terms with the parameters: $\mu = 0.015$ and $\theta = 0.03$. (a) A normal image of a 3D model; (b) the result generated by the current Laplacian-based method; (c) the result generated by the gradient-based method.

3.3 Bas-relief generation and stylization

Given a 3D scene, how to produce different styles of bas-reliefs? As a preliminary solution to this problem, we present two basic styles: round bas-reliefs and flat bas-reliefs. The round style preserves the 3D appearance heavily, and the flat style possesses a global planarity without loss of local fine details. Towards this end, the conceptual formula in Equation (1) is transformed as follows,

$$\min_{H'}(||\mathcal{L}(H') - \phi(\mathbb{L}(\boldsymbol{N}))||^2 + \mu^2 \sum_{i \in C}(h'_i - \theta)^2), \quad (4)$$

where $H' = \{h'_1, h'_2, \cdots, h'_n\}$ is the new resulting height field from the above operation, $\mathcal{L}(H')$ and $\mathbb{L}(\boldsymbol{N})$ are the $Laplacians$ of height field H' and normal image $\boldsymbol{N}$ respectively, θ is a threshold used to control the height of bas-relief, $C = \{i_1, i_2, \cdots, i_s\}$ is the set of indices of the vertices which will be flattened.

The bas-relief heights can be recovered by integrating over the gradients like previous work [6], [9], [8], [10]. Similar results are obtained if gradients are used instead of $Laplacians$. An example is shown in Fig. 5. However, we have used the second order derivative ($Laplacian$) instead of the first-order derivative (gradient) to recover the heights because the former is a scalar, while the latter is a vector, and it is easier to tune the scalar $Laplacian$ using some user-defined functions to control the fine details from orthogonal views. The function $\phi(\cdot)$ in Equation (4) is used to raise or diminish the $Laplacians$. It is flexible in our method and we set $\phi(t) = kt$ for most examples in this paper. The detailed features will be attenuated if $k < 1$ and boosted if $k > 1$ as μ is fixed. If we define function $\phi(\cdot)$ as a low-pass filter, the height field will be smoothed and the detailed features of the resulting bas-relief will be suppressed.

The parameter μ in Equation (4) is used to control the roundness or flatness of the resulting bas-relief. A larger μ will generate a flatter bas-relief. Via this parameter, the designer can designate the bas-relief style easily. Two examples are shown in Fig. 6 and Fig. 7. k is a constant ($k = 1$) for the bas-reliefs shown in Fig. 6. The bas-relief

(a) (b) (c)

Fig. 6: Bas-relief modeling and stylization examples with $\theta = 0.05$ and different values of μ: (a) μ=0.001; (b) μ=0.02; (c) μ=0.2.

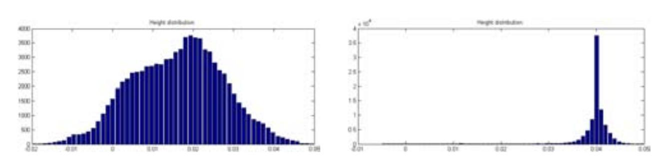
(a) (b) (c)

Fig. 7: Another bas-relief modeling and stylization example with parameters $\theta = 0.05$ and: (a) $\mu = 0.001$, $\phi(t) = t$; (b) $\mu = 0.1$, $\phi(t) = 1.5t$; (c) $\mu = 0.1$, $\phi(t) = 0.1t$.

in Fig. 6(c) is 'flatter' than others, because all height values are crowded to the target value θ when μ is large. Various functions $\phi(t) = kt$ and μ are used in Fig. 7, which produce bas-reliefs with details corresponding significantly different magnitudes.

The 'round' or 'flat' appearance is only a visual effect. In fact, the stylization optimization modifies the distribution of a height field. Fig. 8 shows the distributions of height fields optimized with different parameters. Fig. 8(a) shows the histogram of the height value in Fig. 6(a), and Fig. 8(b) corresponds to Fig. 6(c). As shown in these figures, our stylization optimization turns out a compression or decompression of the height spectrum with fine details preserved.

Furthermore, to attenuate or boost the detailed fea-

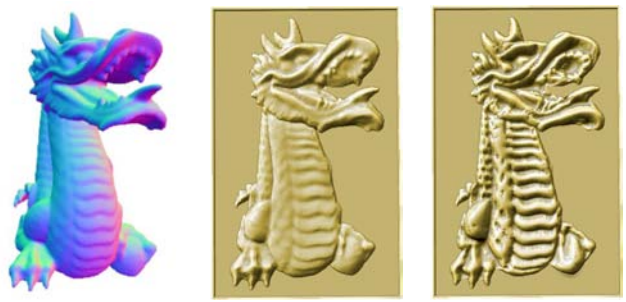


(a) (b)

Fig. 8: Histograms of optimized height fields with different parameters: (a) μ=0.001; and (b) μ=0.2;

(a) (b) (c)

Fig. 9: An example for feature enhancement in a non-linear way. The same parameters μ and θ are set for (b) and (c). (a) a normal image; (b) bas-relief without feature enhancement and (c) bas-relief with feature enhancement.



(a) (b) (c)

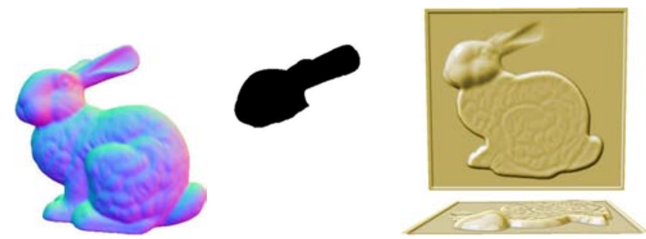
Fig. 10: Another example for feature enhancement. (a) a normal image; (b) bas-relief without feature enhancement and (c) bas-relief with feature enhancement.

tures in a non-linear way, we set

$$\phi(t) = \begin{cases} sign(t) \cdot l, & |t| \geq \tau, \\ sign(t) \cdot l \cdot e^{((|t|-\tau)/\tau)}, & |t| < \tau, \end{cases} \tag{5}$$

where $t = \mathbb{L}(\boldsymbol{N})$, l is positive and τ is a threshold for feature identification. In this formulation, the magnitudes are fixed to a constant for all $Laplacians$ whose absolute magnitudes are above the threshold τ. This function is used to give a suitable scale to the $Laplacians$, making the features distribute in a uniform way. Different from the linear function $\phi(t) = kt$, $\phi(t)$ defined as Equation (5) suppresses the details and noises corresponding to high frequencies and boosts smoothly the details corresponding to low frequencies. For the examples shown in Fig. 9 and Fig. 10, we set $l = 0.5l_m$ and $\tau = l$ where l_m is the maximum absolute magnitude of $Laplacians$ in the normal image. As shown in Fig. 9 and Fig. 10, the use of the above parameters enhances detailed features over the resulting bas-reliefs. The parameters given above are suggested values, and one can adjust them if needed. Increasing l causes the detailed features to be further improved all over the relief shape, while reducing l suppresses fine details.

Hybrid bas-relief: For a monotonic bas-relief style, the indices set C in Equation 4 includes the whole region by default. However, we also define the indices set C to make hybrid bas-reliefs which are partial round and partial flat. An example is shown in Fig. 11, where a mask image is imposed on the Bunny's normal image to indicate the head part should be round and the body part should be flattened. Besides mask images, layers (in section 4) can be used to define indices sets.



(a) (b) (c)

Fig. 11: A hybrid bas-relief example. (a) A normal image; (b) a mask image; and (c) the resulting hybrid bas-relief.

4 Layer-Based Framework

4.1 Normal image layer

Image layers are available in a wide range of image processing programs (e.g. Adobe Photoshop) and even 3D modeling tools (e.g. Autodesk Mudbox) also introduce this feature. We also introduce layers for our bas-relief modeling method. We package the normal image layer as a 4-channel image, which includes three normal channels and one alpha channel. In our current implementation, we directly blend the layers. We linearly combine image layers to blend detailed features from different layers. Each layer has an opacity value which falls in the range $[0, 1]$. For each layer, the background (pixels without normals) is transparent. When the opacity of a layer is 1, this layer occludes the layers beneath it in the non-transparent regions. Given a group of layers, we compute the blended $Laplacians$ according to the opacity values. Finally, we integrate the blended $Laplacians$ only one time to get a blended bas-relief.

An example is shown in Fig. 12. A front layer (angel) blends rather than occludes a back layer, which forms a blended bas-relief. The composite normal image is composed of two layers using the alpha-channels of images ($\alpha = 0.5$).

4.2 Layer-based editing

In our layer-based framework, the normals rather than $Laplacians$ are used as inputs of our algorithm. Besides the intrinsic merit mentioned above, the detailed features of normal images with the blue-purple-ish appearances are more discriminable than the ones of grayscale images which stand for height fields or $Laplacians$.

We can combine two layers without consideration of the alpha values, or setting the alpha to 1 for the foreground. The occlusion between layers is decided by the order of each layer. Note that, the normals of

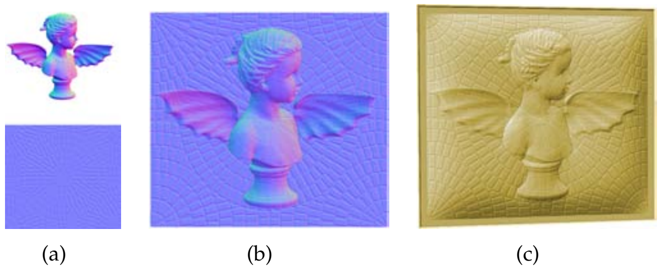


Fig. 12: Layer-based blending of normal image layers: (a) two normal image layers (b) the blend normal image; and (c) the resulting bas-relief.

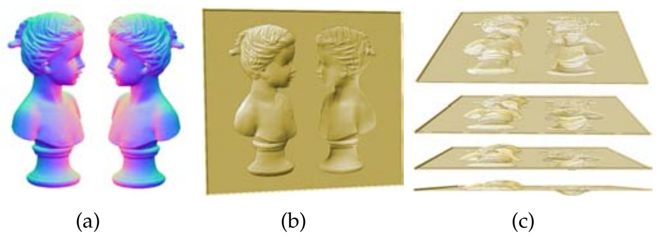


Fig. 13: A bas-relief created using inconsistent normal images: (a) two orientation inconsistent normal images; and (b,c) the resulting bas-relief shown from different views.

all layers should be coded in a consistent coordinate system. One should abide by this rule while editing a layer. For our current implementation, we classify the transformations into two types, one affects the normal orientations such as **flip** and **rotate**, and the others do not, such as **translate** and **scale**. For the first type, if an image is transformed by a matrix T, the normal direction of each pixel should be transformed by the same matrix, $\boldsymbol{N'}(u,v) = T\boldsymbol{N}(u,v)$. Though scale transformations does not change the normal orientation, we should re-normalize the normals after scaling the normal images.

In addition, inconsistent normal images can be used to produce another visual effect. If we combine the inconsistent normal images, a relief with bumps and dents will be generated. An example is shown in Fig. 13, where two inconsistent normal images are put together to result in a bas-relief with raised and sunken angels. The right angel is derived from the left one by using a horizontal flip. The inconsistent orientations of normals are composed to design more kinds of bas-reliefs.

In our layer-based framework, the cut-and-paste operation can be easily used to design new reliefs using existing normal images. Using this operation, one can easily paste parts of a normal image onto another image. An example is shown in Fig. 14. We cut and warped a part of the normal image from the buddha model, and then put it above the elephant layer to form a new design. Another example which includes three layers is shown in Fig. 15. In this example, we demonstrate the **flip** operation. The right wing (see Fig. 15(a)) is cut from the gargo model, and we flip it to generate the left wing with consistent orientation, then merge them to form a consistent bas-relief. With some easy manipulations of images rather than 3D models, we can quickly design an interesting bas-relief.

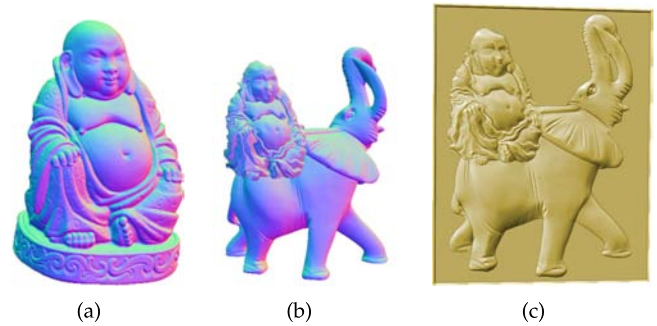


Fig. 14: Normal image editing using cut-and-paste operations: (a) a buddha normal image; (b) a composite normal image by pasting a warped buddha onto an elephant; and (c) the resulting bas-relief.

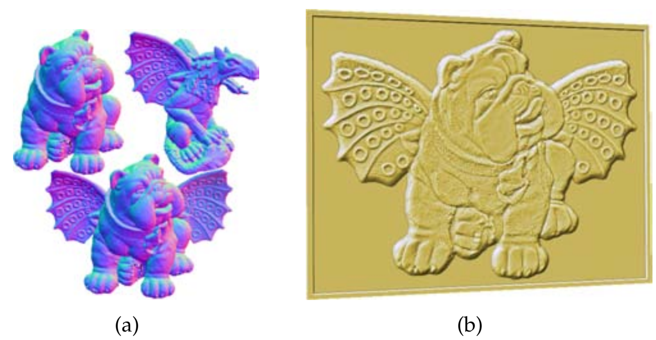


Fig. 15: Bas-relief created by cut-and-paste operations in conjunction with a **flip** transformation of the wing: (a) a composite normal image created by merging the normal images; and (b) the resulting bas-relief.

Furthermore, operations of details peeling and transplanting are also implemented in our framework. For the example in Fig. 1, the dragon squama details are peeled and attached onto a fish model (a smooth normal image). To highlight the effect, we amplify the dragon squamae.

5 Results and Discussions

In this paper, normal images are always described in camera space. The appearance of normal images in camera space almost always looks bluish because of their normal spectrums. The z components of normals in camera space (visible from the camera) are always positive. A camera space normal image is free from mesh parametrization and view-dependent. We first transform a normal vector from object space into camera space and normalize the result. The normalized normal vector components fall in the interval $[-1, 1]$. To store as color channels of an image, we linearly transform the components $\boldsymbol{N}(u,v)_x$, $\boldsymbol{N}(u,v)_y$ and $\boldsymbol{N}(u,v)_z$ into the interval $[0, 1]$, where $\boldsymbol{N}(u,v)$ stands for a normal image. The range-compressed vector is $\boldsymbol{N'}(u,v) = (\boldsymbol{N}(u,v) + (1,1,1))/2$.

(a) (b) (c)

Fig. 16: Bas-reliefs produced by our method with different parameters: (a) a normal image; (b) a round bas-relief for $\mu = 0.001$ in Equation 4; and (c) a flat bas-relief for $\mu = 0.2$.

We then render the final normal vectors into a texture image. We use the OpenGL Shading Language to render camera space normal images in the OpenGL context.

5.1 Height field recovery

The optimization problem defined in the Equation 4 is quadratic and can be reformulated as a sparse linear system

$$A\boldsymbol{X} = \boldsymbol{b}. \tag{6}$$

It can be solved in a least squares sense. Two boundary conditions can be selected to solve this equation. We may apply zero-value constraints at a regular boundary (a rectangle) which leads to a rugged background along the image silhouettes. If we want a flat background, we should apply zero-value constraints at the boundary of a normal image. In the rest of the paper, we apply the latter boundary constraints unless otherwise specified. Equation 6 can be solved using an iterative method, such as Gauss-Seidel solver or multi-grid solver, etc. We use the direct solver of [17] in our implementation.

Our solution of 3D bas-relief modeling and stylization is concept simple and easy to implement. However, there are essential improvements on previous methods. The motivation of our solution is to make bas-reliefs with similar appearance for orthogonal views of the given 3D scenes. And our method is *free of depth discontinuity intrinsically* and without handling the gradients explicitly like previous methods because the gradient is calculated from the normal at each pixel *without involving the adjacent pixels*.

5.2 Experimental results

Our variational approach has a few control parameters which makes it relatively easy to adjust and control. The threshold θ in Equation (4) is set to fix the height range of the resulting bas-relief. In our algorithm, the panel width of bas-relief is set to 1.0 and θ is the ratio between height range and panel width. The parameter μ in Equation (4) is used to maintain a balance between details, or fidelity, and the height constraint. As shown in above figures, a small μ produces fine details with overall 3D structure, while a larger value suppresses the overall structure with more flattened results. To obtain the same resulting height range from various resolutions of normal images for the same scene, we normalize the $Laplacians$ in Equation (2): $\mathbb{L}(\boldsymbol{N}) = Div(\nabla H')/w$ where w is the width of the normal image. As a small μ will relax the height constraint, the final step of our algorithm is simply to linearly scale the resulting heights from the Equation (4) to fit within the final dynamic range. As we use ratio values (1.0 and θ) in the optimization (4), so the threshold parameter μ is easy to control. The examples in Fig. 6, 7 and 16 show the influence of μ which can be regarded as a guide.

Table 1 highlights the parameters of most examples shown in this paper. The parameters listed in this table are suggested values, and one can adjust them intuitively if necessary. However, a default setting of μ is given in order to provide a quick choice with a reasonable trade-off between the preservation of the overall 3D structure and the height constraint. We provide three options: $\mu = 0.001$ for round styles, $\mu = 0.01$ for medium styles and $\mu = 0.1$ for flat styles. The default value is set as $\mu = 0.001$. In addition, once the μ is fixed, we can further emphasize the detailed features by means of the function $\phi(t)$ if needed.

Examples	μ in Equation (4)	θ in Equation (4)
Eight(Fig. 2)	0.001	0.03
Golf(Fig. 9)	0.02	0.1
Dragon(Fig. 10)	0.02	0.06
Bunny(Fig. 11)	0.1	0.03
Blending layers(Fig. 12)	0.005	0.06
Angels(Fig. 13)	0.001	0.05
Cut-and-paste I(Fig. 14)	0.003	0.07
Cut-and-paste II(Fig. 15)	0.08	0.04
Buddha(Fig. 17)	0.001	0.08
Bunny(Fig. 18)	0.001	0.08
Hollow(Fig. 19)	0.015	0.04

TABLE 1: Parameters μ and θ of most examples for bas-relief modeling and stylization shown in the paper. We set $\phi(t) = t$ for all these examples.

As mentioned above, we solve the Equation (6) by the direct solver proposed in [17]. The Cholesky factorization of matrix $A^T A = R^T R$ is first found, where R is an upper triangular matrix. Then $\boldsymbol{X}$ is found by solving two triangular linear systems, that is $R^T\boldsymbol{Y} = A^T\boldsymbol{b}$ and $R\boldsymbol{X} = \boldsymbol{Y}$. As stated in [17], the factorization is fast enough for practical applications. Once the factorization is finished, we can get the result quickly when we alter $\phi(t)$ in Equation 4. Similar to previous methods [6], [9], [8], [10], the most time-consuming step is to solve a sparse linear system (6), so our method is as efficient as most previous work. We implemented our algorithm and tested our examples on a PC with an Intel Core2Duo 1.80 GHz CPU and 1 GB memory using C++ and OpenGL.

One of our goals is to reconstruct the geometry so

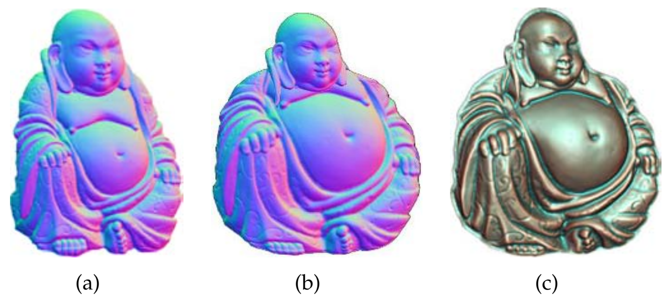


Fig. 17: Bas-relief produced from a warped normal image: (a) a normal image; (b) a warped image of (a) via the 'spherize' distort-filter in Adobe Photoshop; and (c) the resulting bas-relief.

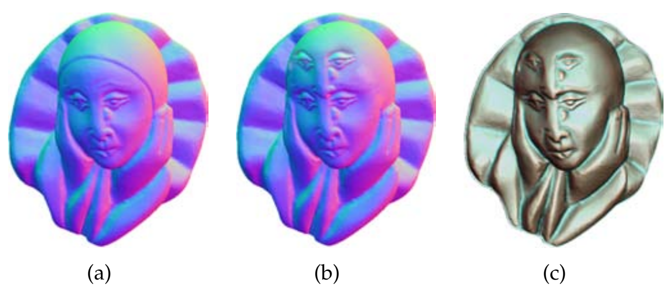


Fig. 18: Bas-relief produced from a normal image modified by Adobe Photoshop: (a) a normal image; (b) a modified version using Adobe Photoshop; and (c) the resulting bas-relief.

that the normal of a resulting bas-relief should appear similar to that of the original scene. In Fig. 16, we can see the bas-relief shown in Fig. 16(b) preserves very fine details with a bigger range of depths, while the bas-relief shown in Fig. 16(c) possesses similar fine details with a smaller range of depths. These different styles of bas-reliefs can be targeted at different environments or different materials. Our algorithm provides somewhat freedom to design bas-reliefs with different styles.

The example in Fig. 17 shows that our method can manipulate normal images edited by any general image software, such as Adobe Photoshop. In this example, we use the 'spherize' distort-filter of Adobe Photoshop to produce a warped bas-relief. Another examples using Adobe Photoshop is shown in Fig. 18.

Fig. 19 shows an example for producing hollow bas-relief using our method. The round hollow bas-relief in Fig. 19 is produced from a composite normal image. This example shows that our method is suitable to produce hollow bas-reliefs.

As mentioned above, our algorithm allows one to select and combine different design styles and introduces bas-relief editing in normal image space. Fig. 20 shows some complex bas-reliefs produced by means of layers. Fig. 20(a) and 20(b) show two interesting bas-reliefs generated by blending two normal image layers. The examples shown in Fig. 20(c) and 20(d) are round bas-reliefs which are created by merging different normal

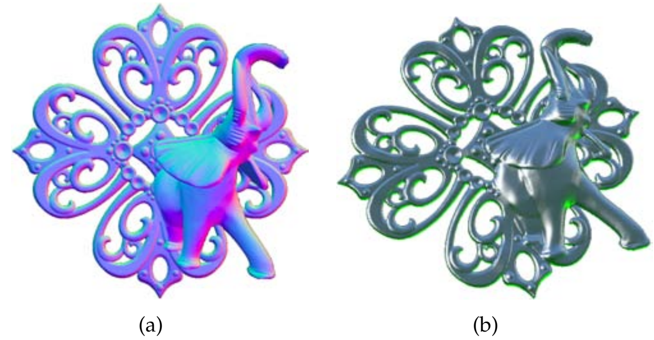


Fig. 19: A hollow bas-relief: (a) a normal image; and (b) the resulting hollow bas-relief generated from (a) using our method.

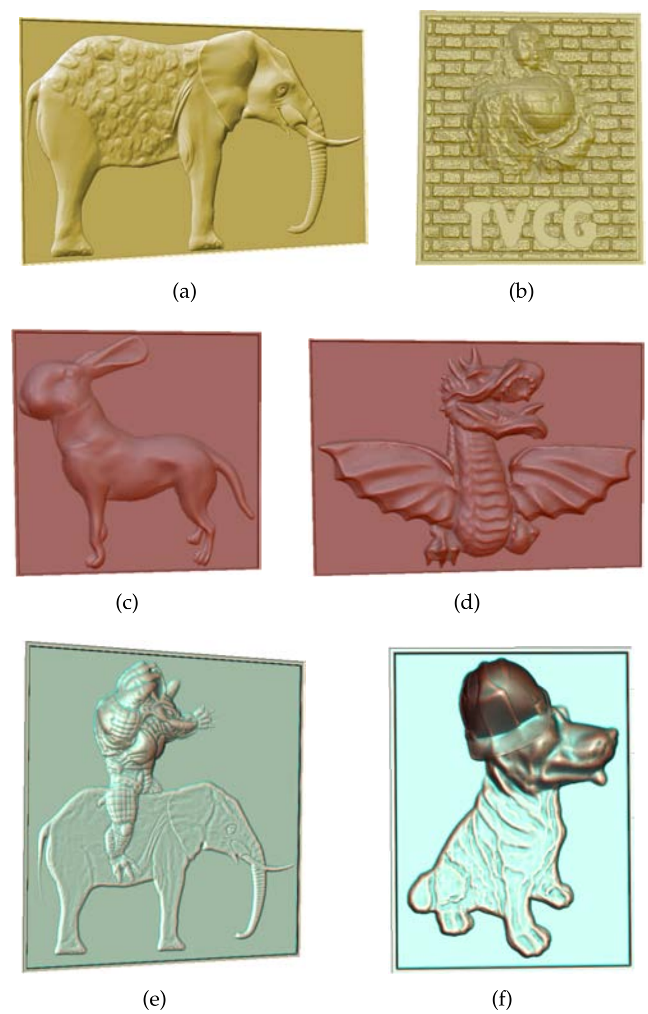


Fig. 20: Some bas-reliefs created using our method.

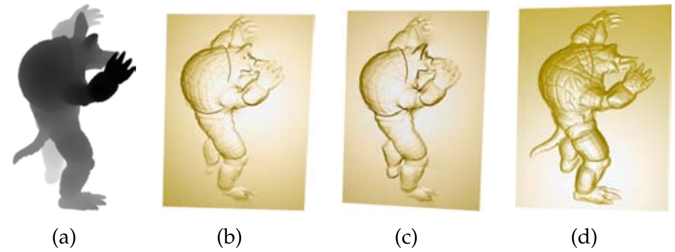


Fig. 21: Comparison with Weyrich's method: (a) depth of 3D model; (b) one result of Weyrich's method; (c) another result of Weyrich's method; and (d) our result.

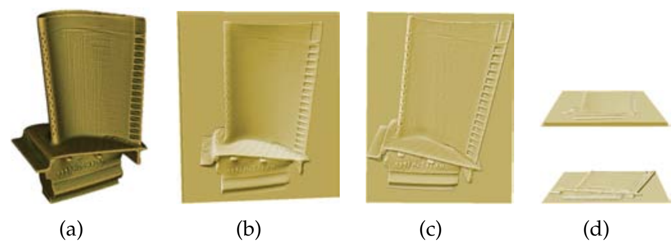


Fig. 22: Comparison with Sun's method: (a) a 3D model; (b) bas-relief produced by our method ($\mu = 0$); (c) bas-relief produced by Sun's method; and (d) comparison in another view.

images. Fig. 20(e) and Fig. 20(f) provide two hybrid bas-reliefs produced using our method.

5.3 Comparisons

In the following, we compare our method with other bas-relief generation methods. We begin by comparing our method with that of Weyrich et al.[6]. Their problem focuses on the creation of a bas-relief from a 3D scene and their solution adapts methods from the tone-mapping literature. An example is shown in Fig. 21. Their method produces sound bas-reliefs but depends on a parameter to eliminate the discontinuity on the occluding boundary.With our method, it automatically guarantees a discontinuity free solution and can produce similar bas-reliefs (the round style) with theirs.

We then compare our method with that of Sun et al.[11]. Bas-reliefs generated by their method are shown in Fig. 22. This model gives a big range of depth especially at the silhouette. Sun's method (see Fig. 22(c)) flattens depth well and preserve most detailed features. Our method produces a round bas-relief. As shown in Fig. 22(d), the boundary of our result is smooth and consecutive with the background, while the one of Sun's method is jaggy with discontinuities. Furthermore, our method preserves the original appearance (the concave middle) and small scale details.

In Fig. 23, we compare our method with the feature preserving methods of Kerber et al.([7] and [8]). We flatten the bas-relief via parameters $\phi(t) = t$ and $\mu = 0.0001$. Our result preserves features as well as theirs.

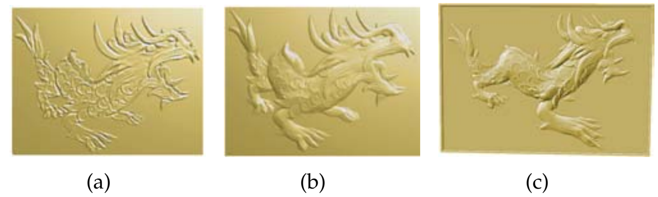


Fig. 23: Comparison with Kerber's methods: (a) one of Kerber's method;(b) another Kerber's method;and (c) our result.

5.4 Other applications

Bas-reliefs modeling using images: Besides 3D models or scenes, normal images can also be created from general images. In fact, we compute the normals from general images easily and straightly. We do not focus on restoring the exact normals corresponding to the underlying fully extended 3D objects in the general image, because the elements like luminance, color and texture in general colored images usually do not reflect the geometric features properly. It manifests problems when using general colored images as input or those which contain complex textures, even using the sophisticated algorithms such as SfS. In our current bas-relief modeling system, general images are just used as 'background patterns' or decorations. We first convert a general colored image into a grayscale image whose luminance is directly regarded as height, and then compute the normal information of the grayscale (height) image straightly as follows,

$$\mathbf{n}(u, v) = (-\nabla I(u, v), 1) = (-\partial I/\partial u, -\partial I/\partial v, 1), \quad (7)$$

where $I(u, v)$ is the luminance of pixel (u, v). We then normalize the vector $\mathbf{n}(u, v)$ to obtain the normals $\mathbf{N}(u, v)$.

A bilateral filter [18] is applied on normals to smooth the normal image while preserving the edge features. An example is shown in Fig. 24. A normal image converted from a texture image blends with two other normal images created from two 3D models. An interesting and hybrid bas-relief model is created from the composite normal image using our method. Fig. 25 shows another example. In our current implementation, general images are integrated into our method to produce 'geometric textures', which provides an interesting tool for bas-relief modeling.

Normal Image to Displacement Map: Normal image is commonly used to further enhance the appearance and details of a low resolution polygon model. Displacement map is an alternative technique using a height map to create geometric features, which gives surfaces a better sense of depth and detail. Fortunately, our work directly accomplishes the transformation from a normal image to a displacement map. We have designed some 3D models shown in Fig. 26 using some displacement maps created by our method. Normal images are converted to bas-reliefs which are used as displacement maps to paste

(a) (b) (c)

Fig. 24: Bas-relief modeling using an image and two 3D models: (a) a texture; (b) a composite normal image; and (c) the resulting bas-relief.

(a) (b) (c)

Fig. 25: Bas-relief modeling using an image and a 3D model: (a) an image and its normal image; (b) a composite normal image; and (c) the resulting bas-relief.

onto a base mesh. Our method makes it possible to easily paste 3D models or scenes onto another 3D model, which is an interesting and useful 3D modeling technique.

6 Conclusion and Future Work

Bas-relief is a prevalent art all over the world throughout the history. However, digital bas-relief is a young research topic in the computer graphics field. This paper presented a novel method for digital bas-relief modeling. Following this method, a height map representing a bas-relief is constructed from a normal image and compressed nonlinearly while preserving or even enhancing features of the image. An image-based framework was proposed to make our bas-relief design tool more flexible and powerful. Many examples showed that our method can generate reasonable bas-reliefs.

(a) (b) (c)

Fig. 26: From normal images to displacement maps.

One advantage of our algorithm is freeing of depth discontinuity intrinsically. However, it might be a problem in some special cases. For instance, because our approach removes the depth discontinuity automatically, we cannot reconstruct a cube when it is placed on a flat plane and the camera is looking down on it vertically. In this case, we may need to slope the vertical edges slightly, so that the edge features can be reconstructed.

Our future work will focus on novel methodologies for constructing normal images. A promising direction is to use grayscale images like pencil drawing figures as inputs. We can compute the normals from pixel information, such as shadows and silhouettes. Another interesting direction is to draw the normals via real-time sketch based drawings. One can draw some strokes on the image plane and indicates the normals in some anchor positions or hatching shadows close to the strokes, and the system will instantly generate the normal image.

Acknowledgements

We would like to thank the anonymous reviewers for their constructive comments. This work is supported by the National Natural Science Foundation of China (61202278) and the Zhejiang Provincial Natural Science Foundation of China (Y1111101 and Q12F020016).

References

[1] M. Wang, J. Kerber, J. Chang, and J. J. Zhang, "Relief stylization from 3d models using featured lines," in *Proceedings of the 27th Spring Conference on Computer Graphics*, 2011, pp. 63–68.

[2] J. Kerber, M. Wang, J. Chang, J. J. Zhang, A. Belyaev, and H.-P. Seidel, "Computer assisted relief generation - a survey," *Computer Graphics Forum*, vol. 31, no. 8, pp. 2363–2377, December 2012.

[3] P. N. Belhumeur, D. J. Kriegman, and A. L. Yuille, "The bas-relief ambiguity," *Int. J. Comput. Vision*, vol. 35, no. 1, pp. 33–44, 1999.

[4] P. Cignoni, C. Montani, and R. Scopigno, "Computer-assisted generation of bas- and high-reliefs," *J. Graph. Tools*, vol. 2, no. 3, pp. 15–28, 1997.

[5] R. Fattal, D. Lischinski, and M. Werman, "Gradient domain high dynamic range compression," in *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, 2002, pp. 249–256.

[6] T. Weyrich, J. Deng, C. Barnes, S. Rusinkiewicz, and A. Finkelstein, "Digital bas-relief from 3d scenes," in *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, 2007, p. 32.

[7] J. Kerber, "Digital art of bas-relief sculpting," Masters thesis, Universität des Saarlandes, August 2007.

[8] J. Kerber, A. Tevs, R. Zayer, A. Belyaev, and H.-P. Seidel, "Feature sensitive bas relief generation," in *IEEE International Conference on Shape Modeling and Applications Proceedings*. Beijing, China: IEEE Computer Society Press, June 2009, pp. 148–154.

[9] W. Song, A. Belyaev, and H.-P. Seidel, "Automatic generation of bas-reliefs from 3d shapes," in *SMI '07: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2007*, 2007, pp. 211–214.

[10] Z. Bian and S.-M. Hu, "Preserving detailed features in digital bas-relief making," *Computer Aided Geometric Design*, vol. 28, no. 4, pp. 245–256, 2011.

[11] X. Sun, P. L. Rosin, R. R. Martin, and F. C. Langbein, "Bas-relief generation using adaptive histogram equalization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 4, pp. 642–653, 2009.

[12] M. Alexa and W. Matusik, "Reliefs as images," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 1–7, 2010.

[13] Z. Li, S. Wang, J. Yu, and K.-L. Ma, "Restoration of brick and stone relief from single rubbing images," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 2, pp. 177–187, 2012.
[14] J. Wu, R. R. Martin, P. L. Rosin, X. Sun, F. C. Langbein, Y.-K. Lai, A. D. Marshall, and Y.-H. Liu, "Making bas-reliefs from photographs of human faces," *Computer-Aided Design*, vol. 45, no. 3, pp. 671–682, 2013.
[15] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah, "Shape from shading: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 690–706, 1999.
[16] T.-P. Wu, C.-K. Tang, M. S. Brown, and H.-Y. Shum, "Shapepalettes: interactive normal transfer via sketching," in *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, 2007, p. 44.
[17] T. Davis and W. Hager, "Dynamic supernodes in sparse cholesky update/downdate and triangular solves," *ACM Trans. Math. Software*, vol. 35, no. 4, 2009.
[18] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of ICCV*, 1998, pp. 839–846.

Zhongping Ji received his BSc degree in Mathematics from Northwestern Polytechnical University in 2003 and PhD degree in Mathematics from Zhejiang University in 2008. He is a lecturer in the School of Computer Science and Technology at Hangzhou Dianzi University. His research interests include digital geometric processing, computer graphics, and image processing. He is a member of Geometric Design and Computing Committee of China Society for Industrial and Applied Mathematics.

Weiyin ma received his BSc and MSc degrees from East China Institute of Technology (ECIT) in 1982 and 1985, respectively, and MEng and PhD degrees from Katholieke Universiteit Leuven (K.U.Leuven) in 1989 and 1994, respectively. He is an associate professor of mechanical engineering at City University of Hong Kong. His present research interests include digital geometry processing, computer aided geometric design, CAD/CAM, isogeometric analysis, 3D printing, rapid prototyping and manufacturing.

Xianfang Sun received his PhD degree in control theory and its applications from the Institute of Automation, Chinese Academy of Sciences. He is a lecturer at Cardiff University. His research interests include computer vision and graphics, pattern recognition and artificial intelligence, and system identification and control. He is on the Editorial Board of Acta Aeronautica et Astronautica Sinica.