Bas-Relief Modeling from Normal Layers

Mingqiang Wei, Yang Tian, Wai-Man Pang, Charlie C. L. Wang, Ming-Yong Pang, Jun Wang, Jing Qin and Pheng-Ann Heng

Abstract—Bas-relief is characterized by its unique presentation of intrinsic shape properties and/or detailed appearance using materials raised up in different degrees above a background. However, many bas-relief modeling methods could not manipulate scene details well. We propose a simple and effective solution for two kinds of bas-relief modeling (i.e., *structure-preserving* and *detail-preserving*) which is different from the prior tone mapping alike methods. Our idea originates from an observation on typical 3D models, which are decomposed into a piecewise smooth *base layer* and a *detail layer* in normal field. Proper manipulation of the two layers contributes to both structure-preserving and detail-preserving bas-relief modeling. We solve the modeling problem in a discrete geometry processing setup that uses normal-based mesh processing as a theoretical foundation. Specifically, using the two-step mesh smoothing mechanism as a bridge, we transfer the bas-relief modeling problem into a discrete space, and solve it in a least-squares manner. Experiments and comparisons to other methods show that (i) geometry details are better preserved in the scenario with high compression ratios, and (ii) structures are clearly preserved without shape distortion and interference from details.

Index Terms—Bas-relief modeling, normal decomposition, detail-preserving, structure-preserving, discrete geometry processing

1 Introduction

Bas-relief is a representative art form that has a long history in many cultures. By now representing bas-reliefs digitally [1], [2], [3], the creation of bas-relief sculptures relies less on the skills and experiences of sculptors [4]. Although many difficulties in the traditional production of bas-reliefs are overcome [5], [6], modeling a 3D scene to a highly compressed bas-relief with either well-preserved details or structures is still challenging. For example, to mimic the manual method of bas-relief production in Fig. 1 is fairly arduous by inputting a 3D scene.

Bas-relief modeling tries to transform 3D geometry into 2.5D reliefed surfaces. It is produced by squeezing a 3D scene consisting of objects along a particular direction. Most bas-relief modeling methods adapt high dynamic range (HDR) compression techniques [7], [8] from the vision community. The input 3D geometry is viewed as a height field for direct compression to a lower dynamic range [9], or it is regarded as a gradient or normal field in compression [10], [11], [4]. Since there is no explicit identification of fine details in these HDR-based methods, compressing the dynamic range commonly prefers the shape of base surfaces, rather than fine details which may constitute smaller areas in a bas-relief.

Fig. 1. Bas-relief sculptures by manual production. The left column shows a very shallow cultural bas-relief. The second shows a large bas-relief set that is about 20 meters high. The fourth shows two mini bas-reliefs and the third shows the zoomed-in fragments of bas-reliefs from the second and fourth columns respectively. The last column shows a manhole cover with a representative building bas-relief on it. The detailed appearances of the first four bas-reliefs are well-preserved, while the intrinsic properties of the last one is well-carved. By adopting an effective technique analogous to image smoothing and enhancement, our method exhibits a more powerful capability to preserve both an element's details and structures in a 3D scene than existing methods when producing bas-reliefs.

We define two types of bas-reliefs, i.e., *detail-preserving* and *structure-preserving* bas-reliefs. The first type is a scene's visible shape and details, which are all reflected on a bas-relief (detail-preserving); the second is to clearly preserve the visible shape, while

M. Wei and J. Wang are with Nanjing University of Aeronautics and Astronautics, China (mingqiang.wei@gmail.com, davis.wjun@gmail.com).
Y. Tian and P.-A. Heng are with The Chinese University of Hong Kong and Guangdong Provincial Key Laboratory of Computer Vision and Virtual Reality Technology, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China (ytian@cse.cuhk.edu.hk, pheng@cse.cuhk.edu.hk).
C. Wang is with Delft University of Technology, the Netherlands (c.c.wang@tudelft.nl).
W.-M. Pang is with Caritas Institute of Higher Education, Hong Kong SAR, China (wmpang@ieee.org).
M.-Y. Pang is with Nanjing Normal University, China (panion@netease.com).
J. Qin is with the Hong Kong Polytechnic University, Hong Kong SAR, China (harry.qin@polyu.edu.hk).
M. Wei and Y. Tian equally contributed to this work.
Corresponding author: J. Wang.

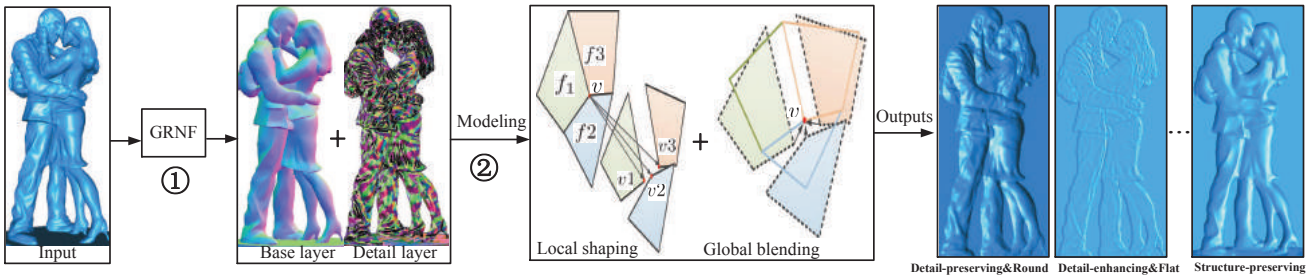


Fig. 2. Our bas-relief modeling method follows the paradigm of normal-based mesh processing, which consists of normal filtering and shape reconstruction. First, we extend the rolling guidance normal filter (RNF) [12] to the GMM-based RNF (GRNF), so that it can decouple the normal field of a mesh to a *base layer* and a *detail layer*. Given the two normal maps, a surface reconstruction scheme with both detail/structure and height constraints is then proposed to generate either detail-preserving or structure-preserving bas-reliefs.

ignoring the details (structure-preserving). However, existing methods could not reproduce these manual works from an input 3D scene. They either lose geometry details on over-compressed bas-reliefs or could not clearly compress these details on them with standard thickness. In this work, we propose a bas-relief modeling method which can preserve the visible shape and/or geometry details of the input model(s).

A variety of mesh smoothing and denoising techniques already exist. In these techniques, isotropic filters are independent to surface geometry which ignore geometric features [13], [14] unless constraints are added [14]. Whereas, anisotropic filters, like bilateral filters [15], [16], mainly focus on eliminating noise introduced by 3D sensing measurement or computational errors whose scales are much less than those of geometric features. They are not intended for, nor do they do a good job of filtering out geometry details-they are designed for removing noise.

We first improve the rolling-guidance normal filter (RNF) [12] to the GMM (Gaussian Mixture Model) based RNF (GRNF) for decoupling the normal field of a mesh to a *base normal field* and a *detail normal field*. Therefore, different from [5], [9], [10], designing a nonlinear compression function to alleviate details lost is not required for our method. The detail normal field generated by the GRNF provides a basis for detail-preserving bas-relief modeling. Meanwhile, the base normal field contributes to structure-preserving bas-relief modeling. Based on the two normal fields, we can construct the mesh of a bas-relief by applying two computation steps: 1) local shaping and 2) global blending, which is free from integrability (see the pipeline in Fig. 2).

The bas-relief modeling problem is actually solved in a discrete geometry processing framework bridged by the well-known two-step mesh smoothing mechanism [17]. To the best of our knowledge, there is no any previous work that applies normal-decomposition-based surface reconstruction to the problem of detail-preserving and structure-preserving bas-relief modeling. The main contributions are three-fold:

- We assume that there are two types of bas-reliefs, one type illustrates both the intrinsic properties and detailed appearance of objects, the other only reflects their intrinsic properties. Correspondingly, we can produce both detail-preserving and structure-preserving bas-reliefs by inputting 3D models.
- We adopt the GMM-based rolling guidance filter to decompose the normal field of a surface to a base layer and a detail layer, and render them independently on the original surface to produce two normal maps. Such an operation not only provides a basis for detail-preserving bas-relief modeling, but also avoids shape distortion when producing structure-preserving bas-reliefs.
- We formulate the bas-relief modeling problem in a discrete geometry processing setup, in order to avoid adding any integrability constraints when recovering the height field. This is different with existing Poisson reconstruction methods.

2 Related Work

There are generally two types of reliefs, i.e., bas-relief and high relief. In contrast to high reliefs [18], [19], in which scene elements are detached from the relief plane, bas-reliefs have elements that are projected into a very narrow depth range [20]. The following survey focuses on bas-relief modeling and normal-based mesh processing techniques.

2.1 Bas-Relief Modeling

Cignoni et al. [21] pioneer the research of bas-relief modeling from an input 3D scene. They have made important observations followed by the subsequent literature. First, the bas-relief modeling problem can be solved over a height field. Second, unused depth intervals at height discontinuities should be removed, guaranteeing a bas-relief to protrude shallowly from the background. From then on, more works focus on 1) preserving the salient features of a reliefed surface

and 2) getting instant feedback in selecting a desirable viewpoint.

First, two types of solutions are usually adopted for preserving salient features. One type notes a similarity to high dynamic range (HDR) imaging, in which the range of intensities of multiple photographs should be compressed in such a way as to display them on an ordinary monitor [7], [8]. For bas-reliefs, depths replace intensities in HDR imaging. Weyrich et al. [10] attenuate gradient discontinuities, while preserving small gradients by using a non-linear compression function, followed by reconstructing a height field by integrating the new gradient field in a least-squares manner. Song et al. [22] work with mesh saliency and shape exaggeration based on the representation of discrete differential coordinates, and a bas-relief is finally generated by a diffusion process. Sun et al. [9] operate compression directly on a height field but use gradient weighted adaptive histogram equalization (AHE) for image enhancement. Ji et al. [11] start from a normal map to reconstruct a bas-relief, instead of a height or gradient field. They can produce quality results with intuitive style control, because normal maps can be freely edited by existing tools, such as Photoshop. They then provide a bas-relief stylization method [23]. Zhang et al. [4] produce a bas-relief by implicitly deforming the original model through gradient manipulation. They later present an adaptive framework for bas-relief generation from 3D objects, with respect to illumination conditions [24]. The other type takes bilateral filter as the main ingredient and increases the proportion of salient features through multi-scale compression functions borrowed from HDR imaging [25], [26]. These methods differ mainly in the compression step, and they can yield impressive results with the salient features preserved. In addition, Schuller et al. [20] use a mesh-based approach to globally optimize a surface that delivers the desired appearance with precise and fine-grained depth/volume control. In summary, bas-relief modeling with structures clearly-preserved, while avoiding shape distortion and interference from details, is not easy for these methods.

In addition to creating a bas-relief from a single object, a recent trend is to bring computational techniques from computer graphics to represent a large 3D scene by a bas-relief set [20], and produce personalized sculptures [27], such as a mini stone with very shallow bas-reliefs on it. In the case of the new challenges, a bas-relief modeling method with each element's details preserving in a 3D scene is more appealing.

Up to here, designing bas-reliefs from input 3D models may be an interactive task, thus, WYSIWYG (what you see is what you get) is more attractive for designers. Many methods, such as Kerber et al.'s [25], Zhang et al.'s [26], and Ji et al.'s [5], are implemented parallel based on modern graphics hardware, that makes real-time artistic design possible for bas-relief modeling.

It is worthy noting that the state-of-the-art methods generate bas-reliefs from natural images [28], [29], [30] and photographs of human faces [31], [32], [33], [34]. However, these methods are often limited due to the fact that color, luminance and texture in an image could not reflect the geometric attributes of objects with complex materials properly.

A discrete geometry processing based method for surface reconstruction has been proposed as Surface-from-Gradients (SfG) [35]. Our method is somewhat related to this work, for both have a fundamental step of recovering height fields over on meshes equipped with surface normals. However, SfG reconstructs a fully 3D object with the proportions of its primitives being the same as in 3D space. Our method is different since it is motivated to construct a height field with a similar appearance of input surfaces under height constraints. Our goal is to achieve the necessary compression without compromising the quality of a model's shape and/or detailed appearance by normal decomposition and surface reconstruction techniques.

2.2 Mesh (Normal) Filter

Normal-based filters of surface meshes were originally designed for mesh smoothing/denoising. Many of these filters have evolved from image denoising techniques, such as from bilateral filters [15], [36], [16], [37], [38] from [39], anisotropic diffusion filters [40], [41], [42] from [43], and L_1/L_0 minimization methods [44], [45] from [46], to name a few. However, adopting these methods for geometry detail removal is non-trivial. Isotropic methods like Laplacian smoothing often lead to shape distortion, and anisotropic methods like bilateral filtering could not effectively remove geometry details. They introduce artifacts during bas-relief modeling.

Recently, a mesh normal filter was proposed as the rolling-guidance normal filter (RNF) [12] by extending the rolling guidance filter [47] in image smoothing. It has shown appealing results in geometry detail removal. By performing the RNF on input meshes and using the Gaussian mixture model (GMM) to fix a decomposition threshold, we can effectively decompose the normal field of an input mesh to a base normal field and a detail normal field.

3 GMM-based Rolling Guidance Normal Filter

The surface decomposition is achieved by the GMM-based rolling-guidance normal filter (GRNF). In the following, we first perform the RNF on the normal field of a mesh to produce a coarse base layer, and then analyze the normal residual by the GMM.

3.1 Rolling Guidance Normal Filtering

Denote a triangular mesh as $M = (V, E, F, N)$, where V, E, F and N are sets of vertex, edge, face and face normal, respectively. The faces in the 1-ring face neighborhood of a face $f_i \in F$ denoted by $N_f(i)$ is the set of faces that have a common vertex or edge with f_i. Denote c_i as the centroid, n_i as the normal and A_i as the area of f_i. The $(k+1)$-th iteration of RNF is defined as [12]:

$$n_i^{k+1} := \bigwedge\Big(\sum_{f_j \in N_f(i)} A_j W_s(||c_i - c_j||) W_r(n_i^k - n_j^k) n_j\Big), \tag{1}$$

where $\bigwedge(\cdot)$ is a vector normalization operator, and $n_i^0 = 0$ for all mesh faces. Both W_s and W_r are Gaussian functions with standard deviations σ_s and σ_r, respectively.

We observe that from the RNF, in the first iteration, n^0 is set to be zero, which regards RNF as a Gaussian filter. Thus, the features whose scales are smaller than σ_s can be filtered out, once σ_s is fixed. Meanwhile, the blurred features with scales larger than σ_s are recovered gradually in the following iterations.

However, 3D objects are often represented by boundary surface meshes without semantic information to describe the base surface and details separately [48]. That is a fact when scanning their corresponding physical objects or when creating them using modeling tools. It means that the complete set of a normal field is involved in the normal filtering, including both the detail layer and the also the base layer. Given a detail-rich mesh M equipped with the face normal field N as input, we can only obtain a coarse base normal field N_B by the RNF. This is because that some structural elements in the base layer are also filtered out to the residuals (see Fig. 3 for an example). We introduce the GRNF in the following subsection to retain the missed parts of the RNF from the residuals for obtaining a holistic base layer.

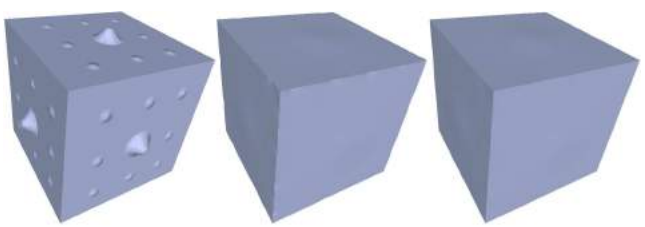


Fig. 3. The sharp edges of the base layer drift due to the information lost during the rolling guidance normal filtering procedure. Whereas, the GMM can take the information back, which demonstrates a better structure-preserving smoothing result. From the left column to the right: The input mesh, the reconstruction results by using the RNF and the GRNF, respectively.

3.2 GMM-based solution

We first perform the RNF to obtain a coarse base layer, called N_B^c. We then obtain a coarse detail layer, called N_D^c (actually the residuals), by subtracting N_B^c from N. N_D^c contains information of N_B which can be further separated. That is, we segment N_D^c into two disjoint components: A detail layer, called N_D and a residual layer, called N_B^r, where $N_B^r + N_B^c = N_B$. This is performed by using a threshold on the vector length vl of each element of N_D^c. We consider that N_D has larger vector lengths than a threshold θ_N above N_B^r. θ_N is automatically determined: We examine the histogram of the vector lengths and approximate it with a GMM with two Gaussians $f = \sum_{i=1}^{2} \alpha_i G(\mu_i, \sigma_i)$, where μ_i and σ_i are the mean and standard deviations of the i-th component in the Gaussian mixture, α_i is its weight with $\alpha_1 + \alpha_2 = 1$. The parameters are estimated using the Expectation Minimization (EM) method [49]. We select the threshold θ_N as the intersection of the two distributions: $\theta_N = \{vh | G_1(vh) = G_2(vh)\}$. Finally, we can obtain a pair (N_B, N_D) with $N_B \cup N_D = N$ and $N_B \cap N_D = \emptyset$. To improve the clarity of the relationship of these symbols, we render them on a real model, as shown in Fig. 4.

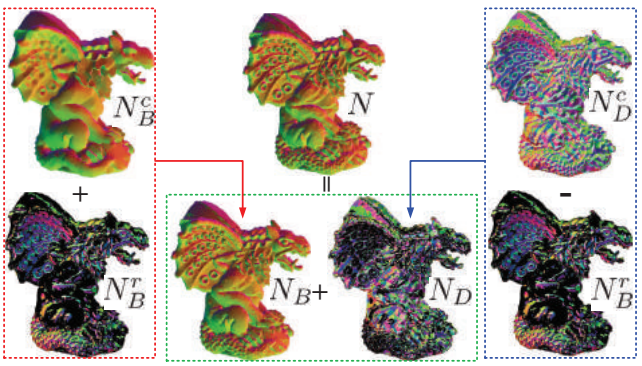


Fig. 4. The relationship among the defined symbols.

In addition, the automatically fixed threshold θ_N is robust to the parameters selection used in the RNF. That means we can loosely select some large values for the parameters of RNF, e.g., $\sigma_r = 0.5$, $\sigma_s = 8\bar{l}_e$ ($\bar{l}_e$ is the average edge length of the input mesh), and the iteration number $k = 5$, and use the GMM to decompose the base layer and the detail layer. Fig. 5 shows the statistical results by the GMM: In the first column, the GMM performs on the coarse detail layer N_D^c using the recommended parameter values of RNF ($\sigma_r = 0.7$, $\sigma_s = 3\bar{l}_e$, $k = 6$); in the second column, the GMM is enforced using the loosed parameter values ($\sigma_r = 0.5$, $\sigma_s = 8\bar{l}_e$, $k = 5$). Due to the fact that the GMM performs on the same bunny model for the first and second columns and both the results have no obvious differences, users can feel free to use our recommended parameter values without any complicated adjustments. The last column shows the effectiveness of the loosed parameter values on

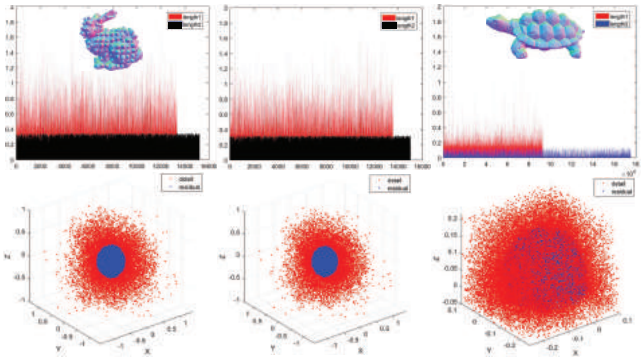


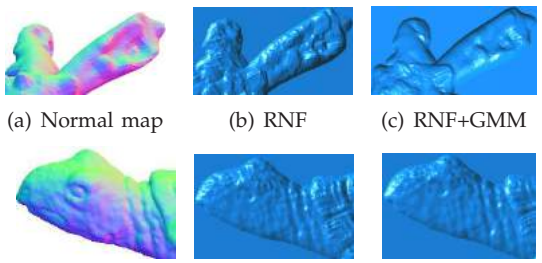
Fig. 5. GMM is robust to the parameters selection in the RNF. The top row shows the vector lengths of the detail normal set N_D (red) and the residual normal set N_B^r (black/blue); the bottom row plots N_D (red) and N_B^r (blue) as dots. For the top row, the vertical axis represents the lengths of the normals, and the horizontal denotes the id numbers of the detail and residual normals. For the bottom row, the x, the y and the z axes represent the x-, the y- and the z-component of a normal respectively (for the better visualization, the third picture keeps the axis with the range of $[-0.2, 0.1]$). The first and second columns show the results on the same bunny model but with a different parameter setting in RNF. The third column shows the results using our loosed parameter values.

another model (we test on a variety of models and they all work well).

The GMM is useful for both structure-preserving and detail-preserving bas-relief modeling. It does not lose information of the base surface in normal filtering for structure-preserving bas-relief modeling; it also does not add false details in normal filtering for detail-preserving bas-relief modeling. As shown in Fig. 6, using the RNF solely leads to the base shape distortion while the GRNF would not; in Fig. 7, using the RNF solely will magnify the base surface, while using the scheme of RNF plus GMM can preserve the real details effectively.

Data-driven RNF. In addition to using the loosed parameters in the RNF, we introduce the data-driven RNF (DRNF) which makes the RNF parameter-free. The DRNF is inspired by the cascaded normal regression (CNR) [50], where non-linear regression functions are modeled by mapping the filtered face normal descriptor (FND) extracted from the neighborhood of a noisy mesh face to the noise-free mesh face normal, and use the modeled functions to compute new face normals based on the two-stage denoising framework [38]. Due to the fact that there are different levels of noise to be removed, multiple iterations of mesh denoising are required, which forms the CNR. Similarly, the DRNF is formulated by the CNR in a normal field at an offline training stage, and is performed at a runtime filtering stage.

Fig. 6. Using RNF plus GMM can prevent shape distortion of the base surface. The first row shows the original normal map (the first one), the base normal maps obtained by the RNF (the second two) and the GRNF (the last two) respectively; the second row shows the original normal map (the first one) and the bas-relief modeling results from their upper counterparts. Compared to the results by the RNF, the results by the GRNF can better preserve the base's structures.



(a) Normal map (b) RNF (c) RNF+GMM

(d) Normal map (e) RNF (f) RNF+GMM

Fig. 7. Using RNF plus GMM can prevent the residual normals N_B^r induced from the base surface.

- **Offline training.** We first perform the GMM-based RNF on a set of original meshes (see Fig. 8) to obtain their base normal fields. We adopt *filtered face normal descriptor* (FND) as the feature descriptor based on the RNF (in the training stage, the set of σ_{s_j} is $\{3\overline{l}_e, 5\overline{l}_e, 8\overline{l}_e, 10\overline{l}_e, 14\overline{l}_e\}$ and the set of σ_{r_j} is $\{0.2, 0.5, 0.8\}$), and formulate a base face normal $\mathbf{n}_i$ as a function $\aleph$ of the original face's FND $\mathbf{S}_i$: $\mathbf{n}_i = \aleph(\mathbf{S}_i)$. We learn $\aleph$ from the FNDs extracted for all base faces and their original normals by neural network, i.e., $\aleph : \mathbf{S}_i \longmapsto \mathbf{n}_i, \forall i$. Multiple iterations are required to reduce the approximation error, where the first regression function coarsely finds the correspondence from the FNDs to the base face normals, and extracts their FNDs to feed into the next regression iteration for a finer approximation.
- **Runtime filtering.** For an input mesh, the learned CNR model consisting of $\aleph$s is enforced on the extracted FNDs to obtain its new face normals. We finally use the GMM to obtain the base and detail normal layers.

We now analyze the reason why we perform surface decomposition in the normal field. First, we should avoid depth discontinuities on occluding boundaries,

Fig. 8. Training data containing twenty-three meshes. The meshes in the first two rows from [50] are enforced by the GMM-based RNF (use the loosed parameters) to obtain their base surface normals; whereas the meshes in the bottom row from [12] are enforced by GMM-based RNF, but use the recommended parameter values therein.

where two neighbor pixels are sampled from two separate triangles during bas-relief modeling. Unlike image gradients that need a forward/backward difference of at least two pixels, the normal at each pixel on a normal map is determined only by its associated triangle. A benefit of processing in the normal field is that it is unnecessary to continue to remove depth intervals at height discontinuities explicitly (see Fig. 9 for an illustration). Second, being faithful to the overall shape of the original mesh, we render the base and detail normal fields on the original mesh to obtain the two normal maps, while not truly updating the mesh vertices to match the two decomposed normal fields and obtaining the two maps on the updated mesh. We can see from Fig. 10, the scheme of rendering the base normal filed on the original mesh leads to less shape distortion than on the updated mesh.

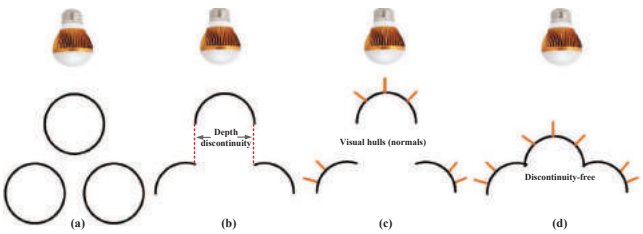


Fig. 9. From a fixed viewpoint, the normals of an input scene (e.g., the three overlapped circles(a)) are calculated independently for each pixel (refer to [5], [23]). As a result, consistent visual hulls equipped by surface normals (c) are produced that can shift the depth discontinuity (b) into a discontinuity-free shape (d).

4 Bas-Relief Modeling

The surface-from-gradients (SfG) was initially used to recover a full 3D surface from captured normal

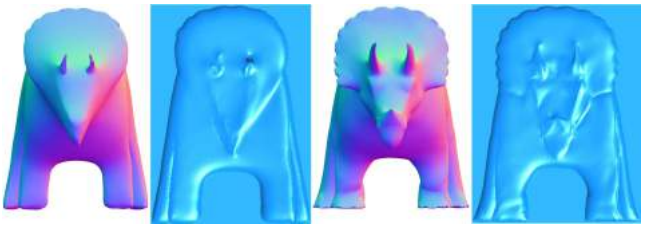


Fig. 10. Rendering the base surface normal field on the original mesh (the third one) can better preserve the overall shape of an object when generating the corresponding bas-relief (the last one). If we really update a mesh to match the base normal field to obtain a base mesh [12] and render it (the first one), the shape distortion would happen when performing bas-relief modeling (the second one)

maps. In the following, we extend the SfG to the scenario of normal-based mesh processing first, i.e., we generalize it to update triangular mesh vertices for matching a filtered normal field, and then propose to use this mechanism to produce a compressed height field (represented by mesh) by inputting one/two normal layers.

4.1 Generalized Surface-from-Gradients

We focus on generalizing the surface-from-gradients (SfG) method [35], which fits our bas-relief modeling well. The generalized SfG (gSfG), in each iteration, determines the position and orientation of each face, according to its filtered normal and current shape, by performing a local shaping step first. Since the mesh is disconnected after local shaping, we glue (stitch) all faces together to a connected mesh by performing a global blending step then, as shown in Fig. 11: In the top row, given a triangular mesh assembled with filtered faces' normals, we first project mesh vertices to their new base planes (a vertex may belong to several faces, thus several corresponding base planes exist) which breaks the mesh, and we then stitch the mesh together into a connected mesh again. The bottom row shows the iteration of performing the two steps. In order to simplify the problem, here we pretend that just the vertex v is moved and all other vertices remain fixed in the bottom row. In local shaping, v is broken into v_1, v_2 and v_3, because it belongs to three faces. In global blending, we glue the broken faces together to form a new mesh (the triangles consisting of dashed line segments mean the new triangles). Because gSfG is a least-squares optimization problem, iteration is often required.

In the local shaping step, the vertices v_1, v_2, v_3 of a face f_i are projected onto its base plane: This plane passes through the centroid c_i of facet f_i and has a new normal n_i'. The projected vertices are denoted by v_1', v_2', and v_3', respectively, as shown in Fig. 12. After simple derivation, without loss of generality, we can

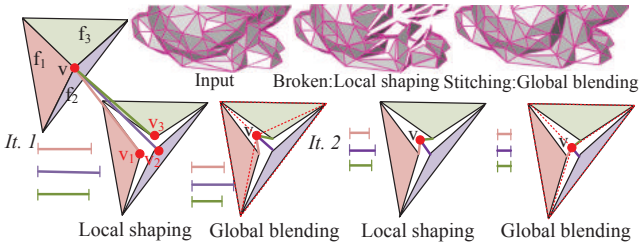


Fig. 11. Illustration of SfG [35] by giving results from real data (top row) and schematic diagrams (bottom row): local shaping and global blending. In the bottom row, the bar means the distance of the vertex v to the projections v_i $(i = 1, 2, 3)$ onto the respective base plane. Local shaping computes the projections using the current estimate v, and global blending updates v by minimizing the sum of the error bars of local shaping, keeping the projections fixed.

obtain the new position v_1' of v_1 as

$$v_1' = v_1 - ((v_1 - c_i) \cdot n_i') \cdot n_i', \tag{2}$$

where n_i' has been normalized.

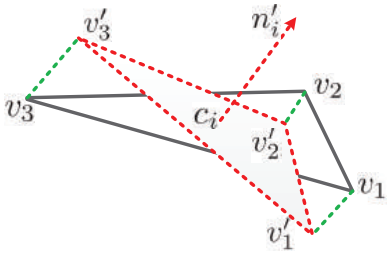


Fig. 12. Illustration of vertex projection. The triangle formed by dashed line segments is the new triangle.

In the global blending step, we aim to deform the mesh into a shape, so that the positions of the vertices of each face give the same shape as the projected face. However, in general, there are several faces adjacent to each vertex v, leading to several projection points. They must be normalized to get a joined-up surface. We use a least-squares optimization in this step. The global blending is formulated as

$$\arg\min \sum_{f_i} \sum_{j=1,2,3} \|v_j - v_j'\|^2, \tag{3}$$

which can be solved by a linear solver.

In order to demonstrate the effectiveness of the gSfG method in shape reconstruction, we employ the local-shaping- and global-blending-based scheme to reconstruct a surface mesh, after its normal field is filtered by the GRNF. Fig. 13 shows the comparison between the Poission reconstruction scheme and our gSfG method. For the two methods, we use the same filtered normal field by the GRNF as the target normal field. We show the smoothed results in the second column. In the third column, we evaluate the deviations between the two smoothed meshes and the original mesh, respectively, by aligning their centroids to minimize the ℓ_2 norm of their vertex deviations. In the fourth column, the vertex deviations (vertex-to-vertex based) between the smoothed mesh and the original mesh are visualized via color coding. Finally, we compute their Hausdorff distances from the smoothed mesh to the original mesh, and the deviations between the smoothed face normals and the target normals, and visualize their distributions using the histogram in the fifth column. Compared to the Poission reconstruction scheme [12], our gSfG method can reconstruct a new mesh that is consistent with the target normal field, while being close to the model's original shape.

4.2 Geometry Shape-Up

In structure-texture decomposition of images [46], an image is decomposed as $I = S + T$, where I, S and T represent the input image, the structure layer and the texture layer, respectively. Therefore, image enhancement can be achieved by manipulating the texture contrast, i.e., $I' = I + \mu \cdot T$ with a user-specified parameter $\mu > 0.0$. Similarly in 3D scenarios, we assume that the underlying surface of an input mesh consists of piecewise smooth patches and details exist within each patch, and formulate the decomposition problem as $N_O = N_B + N_D$ in normal field, where N_O, N_B and N_D are the normal fields of input model, base surface, and details, respectively. Therefore, we can create a height field H, whose details are similar to that of N_O for the orthogonal view by solving [11]

$$\min_{H'} \int\int (\beta \cdot F(H', \delta) + D(N_O + \mu \cdot N_D, H'))dudv, \tag{4}$$

where the compression function $F(\cdot)$ is used to produce a bas-relief with the height of δ, and β controls the style of the bas-relief. In this work, we consider two styles of bas-reliefs: 1) roundness makes the middle portion of a bas-relief elevate the most; 2) flatness makes the prominent part of a bas-relief nearly on a plane. The function $D(\cdot)$ measures the geometric similarity of N_O and H' under the compression, and $\mu > 0.0$ compensates the bas-relief for details lost,.

In the following, we first initialize a background represented by a quadrangular mesh, and then raise its components up in different degrees to convey the detailed appearance and intrinsic shape properties of a bas-relief. To avoid integration, we solve Eq. 4 by two steps: Local shaping and global blending.

4.2.1 *Discrete Geometry Setup*

We use the OpenGL Shading language to transform the original normal field and the detail normal field into eye space, and normalize and render them from an arbitrary view into two textures as two normal maps (images). We call the two normal maps O

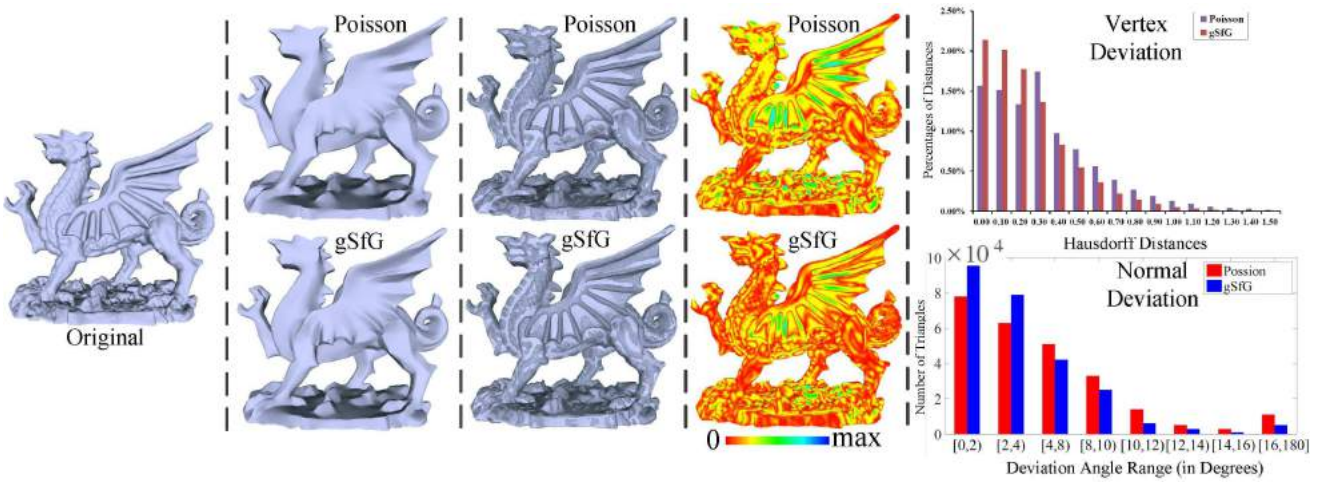


Fig. 13. Comparison of the shape reconstruction between our gSfG and the Poission reconstruction scheme [12], based on the same model and the target normals. The second column shows the smoothed results by the two methods respectively. In the third column, the centroid of each smoothed mesh (flat shading) is aligned with the centroid of the original mesh (flat lines shading), to minimize the ℓ_2 norm of the vertex deviations, and in the fourth column, the vertex deviations are visualized by color coding. It shows that our method leads to the smaller deviation from the original mesh than the Poisson reconstruction scheme. In last column, we compute their Hausdorff distances from the smoothed mesh to the original mesh, and the deviations between the smoothed face normals and the target normals, and visualize their distributions using the histogram. Both the vertex deviation and the normal deviation show that our gSfG method can obtain a smoothed mesh that is consistent with the target normal field, while being close to the model's original shape.

(means Original) and D (means Detail). Because the ranges of components of normalized normals fall into the interval $[-1, 1]$, we linearly transform the normals into the interval $[0, 1]$ to display the normal maps on the screen, i.e., $N'_O = (N_O + (1, 1, 1))/2$, and $N'_D = (N_D + (1, 1, 1))/2$.

We then generate our discrete geometry setup as follows:

• For each two-tuple $< O_{i,j}, D_{i,j} >$, a quadrangular face $f_{i,j}$ is constructed for it. The four vertices of $f_{i,j}$ are $v_{i,j}, v_{i+1,j}, v_{i+1,j+1}, v_{i,j+1}$ respectively; all the faces like $f_{i,j}$ form a quadrangular mesh M. Fig. 14 (left) shows a planar quadrangular mesh which is constructed on a normal image.

• Each vertex $v_{i,j}$ is originally positioned at $((i - \frac{1}{2})l, (j - \frac{1}{2})l, 0)$ with the height field $z = 0$, where l is the edge length of the quadrangular face. The initial quadrangular mesh M is planar. We will show how to drag it to form different bas-reliefs (see Fig. 14 (middle) and (right)).

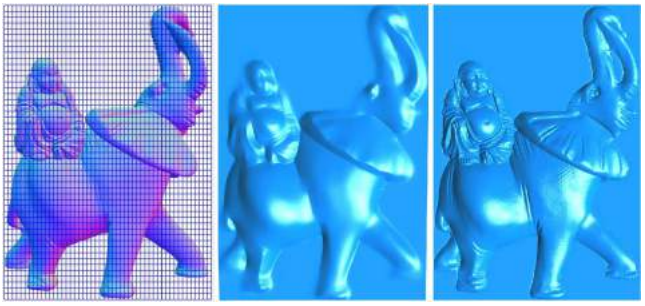


Fig. 14. The planar quadrangular mesh is constructed for a normal image (left), and two round bas-reliefs are produced from this planar mesh with a structure-preserving scheme (middle) and a detail-preserving scheme (right).

4.2.2 Local Shaping

In this step, the vertices of a face $f_{i,j}$ are projected onto two planes O_p and D_p with normals $O_{i,j}$ and $D_{i,j}$ respectively, where the planes pass the current centroid $c_{i,j}$ of $f_{i,j}$. For bas-relief modeling, the x-component and y-component of each vertex are fixed. It is just required to move the vertex along the z axis onto the two planes. The projections of a vertex $v_{m,n}$ along the z axis onto O_p and D_p can be represented by $p^o_{i,j}(v_{m,n})$ and $p^d_{i,j}(v_{m,n})$ respectively, where $m \in \{i, i+1\}$, $n \in \{j, j+1\}$.

A vertex in the quadrangular mesh M is surrounded by four faces. Therefore, a vertex after projections has eight new positions. The z-components of the four vertices in each face can be grouped into two vectors:
$P^o_{i,j} = (p^o_{i,j}(v_{i,j}), p^o_{i,j}(v_{i+1,j}), p^o_{i,j}(v_{i+1,j+1}), p^o_{i,j}(v_{i,j+1}))^T$, and
$P^d_{i,j} = (p^d_{i,j}(v_{i,j}), p^d_{i,j}(v_{i+1,j}), p^d_{i,j}(v_{i+1,j+1}), p^d_{i,j}(v_{i,j+1}))^T$.

4.2.3 Global Blending

After the local shaping step, M is broken into a set of disconnected parts, which need to be glued back

into a connected surface mesh. Here we group the z components of a face $f_{i,j}$ into a column vector:

$$z(f_{i,j}) = (v^z_{i,j}, v^z_{i+1,j}, v^z_{i+1,j+1}, v^z_{i,j+1})^T. \quad (5)$$

On an optimal shape of the quadrangular mesh M, the vectors $z(f_{i,j})$ and $P^o_{i,j}$ should represent the same shape of a face. However, enforcing $z(f_{i,j}) = P^o_{i,j}$ is too restrictive for bas-relief modeling, $z(f_{i,j})$ and $P^o_{i,j}$ are only expected to represent the same shape instead of coincident. Similar to Xie et al. [35], we compare their relative vectors by subtracting the mean of vertices inside $z(f_{i,j})$ and $P^o_{i,j}$ for fast convergence (in our experiments, just one iteration is required).

Moreover, the $P^o_{i,j}$ in Eq. 5 is full 3D. For the purpose of bas-relief modeling, it should be compressed under a depth constraint.

Detail-preserving bas-relief modeling. The generated bas-relief can have the 3D basic structures as similar as possible to the original model, and have well-preserved, even enhanced fine details. Mathematically, we achieve this goal by minimizing the following energy functional

$$\begin{aligned}\Theta(\{v^z_{m,n}\}) = \sum_{f_{i,j}} [\|Rz(f_{i,j}) - RP^o(f_{i,j})\|^2 + \\ \mu\|Rz(f_{i,j}) - RP^d(f_{i,j})\|^2 + \\ \beta\|v^z_{m,n} - \delta\|^2],\end{aligned} \quad (6)$$

where the weight μ is used to preserve the geometry details. The weight β is used to control the style (roundness or flatness) of the resulting bas-relief. The parameter δ is a threshold used to control the height of a bas-relief. R is a matrix

$$R = I_{4\times4} - \frac{1}{4}\mathbf{1}, \quad (7)$$

where $\mathbf{1}$ is a 4×4 matrix with elements all equal to 1. Fig. 15 shows that proper manipulation of the two normal maps together, can provide a basis for meeting the needs of detail-rich bas-relief modeling.

Structure-preserving bas-relief modeling. Another difference with existing methods is that, we can produce the bas-relief which just maintains the overall shape, while ignoring details from a detail-rich model. To this end, we render the base normal field on the original mesh to get the normal map B (means Base) and the corresponding mesh vertices P^b first, and minimize the energy functional

$$\begin{aligned}\Theta(\{v^z_{m,n}\}) = \sum_{f_{i,j}} [\|Rz(f_{i,j}) - RP^b(f_{i,j})\|^2] + \\ \beta\|v^z_{m,n} - \delta\|^2].\end{aligned} \quad (8)$$

5 Results and Analysis

Our prototype has been implemented using C++ and OpenGL. It uses TAUCS as the linear solver. For detail-preserving bas-relief modeling, a sparse and linear equation system with a size of around $1000 \times 900 \times 3$ should be solved (the resolution of each normal map used in this work is fixed with 1000×900; the energy function in Eq. 6 has three terms, thus, the size of the system is triple of 1000×900); while for structure-preserving bas-relief modeling, the system with a size of around $1000 \times 900 \times 2$ is involved in Eq. 8. The experiments are performed on a PC with a 4.00 GHz Intel core i7 and 32 GB of RAM. A user-friendly GUI is created for encapsulating the implemented method and made publicly available to the research community (https://github.com/ivan-hsianglin-kuo/BasReliefModeling). We test our method on individual models with abundant details, a 3D scene of a set of models, and synthesized normal maps to demonstrate the effectiveness of our method.

Parameters. Our method consists of two parts: normal decomposition and normal-based bas-relief modeling. In the first stage, there are three parameters, σ_s, σ_r and k, which are loosely fixed as $\sigma_r = 0.5$, $\sigma_s = 8\bar{l}_e$, $k = 5$ for 3D models. In the second stage, detail preservation, targeted height and stylization for bas-relief modeling are achieved simultaneously by solving Eq. 6. We know from Eq. 6 that there are three parameters which control generated bas-reliefs. We introduce how to tune them: A larger μ can preserve surface details more clearly, and even enhance them. A smaller β keeps the round style of a model which heavily preserves the 3D appearance, whereas a larger β will generate a flatter bas-relief. The height of the bas-relief is controlled by the parameter δ. Experimental results show that $\mu \in [0.01, 1]$, $\beta \in [0.01, 1]$ and $\delta \in [0.1, 30]$ can produce desirable results. Each row in Fig. 16 shows the modeling results by increasing one of the three parameters, when the other two are fixed.

5.1 Normal-Based Reconstruction

After normal decomposition, we render the corresponding 3D model using the two normal fields into two textures using the OpenGL Shading language. This is different from some existing works that need to compute the gradient vector using separate triangles. We do not adopt gradient fields as inputs. In our case, in each normal map, the normal vector of each pixel corresponds to only one triangle on the rendered 3D model. Since we create bas-reliefs from normal maps directly, our bas-relief modeling method is intrinsically free from depth discontinuity, e.g., depth intervals between back objects and front objects. Therefore, we avoid explicitly detecting these discontinuity regions and remove unused depth intervals at these regions. Fig. 17 shows that depth discontinuities at the overlapped regions are eliminated very well with no special processing in advance (they either manually or automatically detect them).

Our method uses normal maps as inputs. It is, therefore, possible to use image editing tools, such as

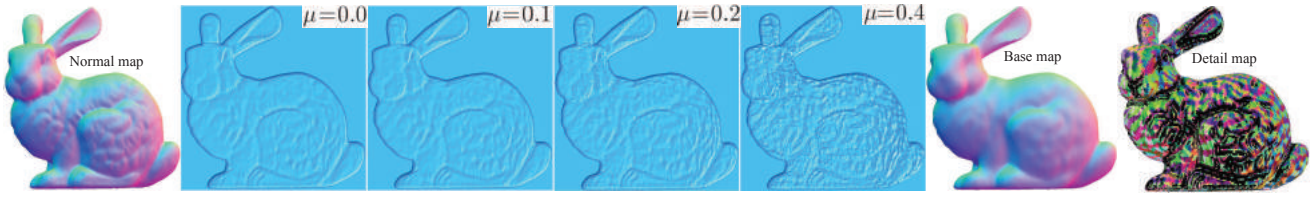


Fig. 15. Our bas-relief results with the original normal map as an input (i.e., $\mu = 0.0$), and with the two normal maps (one is the original map and the other is the detail map with the increasing values of μ) as inputs. Compared to the original geometry (the leftmost one), our bas-relief modeling scheme with the two normal maps outperforms the one normal map in detail preservation. In addition, we can obtain the detail-enhanced bas-reliefs by increasing the detail factor μ, see the fourth and fifth images from the left column. The rightmost two images show the base and the detail maps respectively.

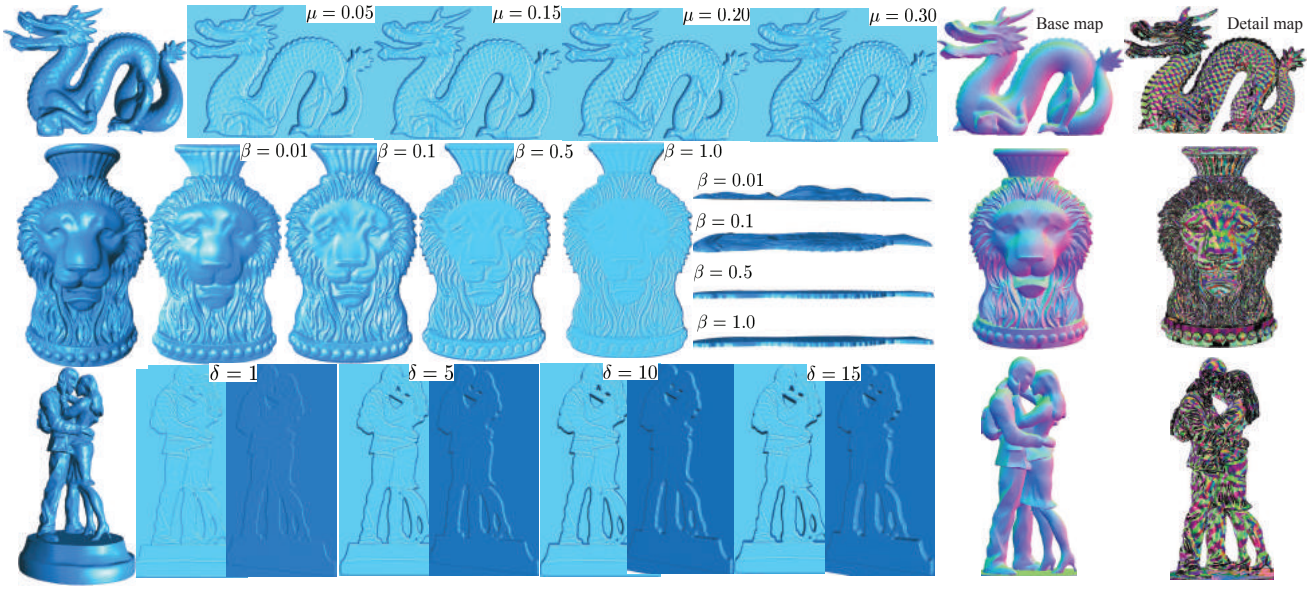


Fig. 16. Our bas-relief modeling results with different parameter values. From the first row, we know that increasing μ can get details more and more clearly ($\beta = 0.3, \delta = 10$). The second row shows that increasing β would lead to flatter results ($\mu = 0.15, \delta = 12$). The third row shows that increasing δ would produce results with higher heights ($\mu = 0.01, \beta = 1$). The last two columns show the base and the detail maps respectively.

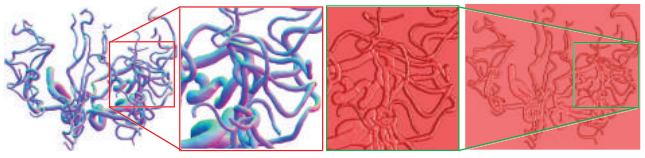


Fig. 17. Our bas-relief modeling on a vascular tree with $\mu = 0.01, \beta = 0.8, \delta = 1$. The method is free from depth discontinuity. The unused depth intervals on occluding boundaries are naturally removed. The left is the 3D model, the right is the modelling result, and the middle are the corresponding zoomed-in fragments.

Photoshop, to freely design bas-reliefs in image space rather than in object space. For instance, two or more layers of normal maps can be synthetized into one normal map, as shown in Fig. 18: The left shows a bas-relief of target ups and downs along the background; the middle shows a bas-relief pasted by details from other objects; the right shows a bas-relief with two models moved together. More examples from normal maps are shown in Fig. 19.

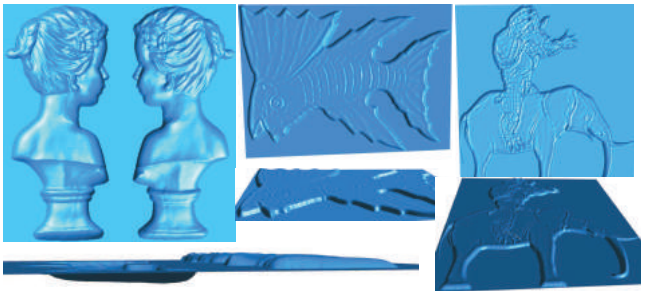


Fig. 18. Bas-relief modeling results for composite normal maps. The bottom row shows the heights of these bas-reliefs from other viewpoints.

In addition, our method can flexibly produce bas-reliefs on curved surfaces. The normal map of the

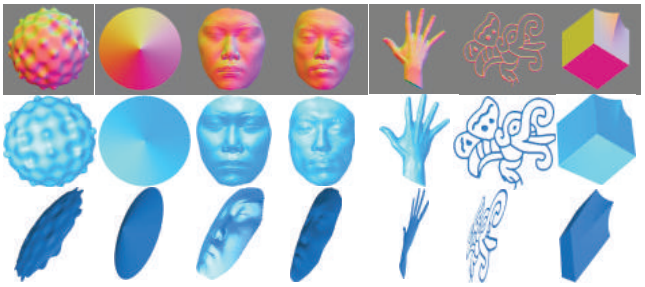


Fig. 19. More results from normal maps. These bas-reliefs have arbitrary boundaries, thanks to the mesh representation. The top row shows the normal maps, and the middle and bottom row gives the corresponding bas-relief modeling results from two viewpoints respectively.

curved surface and the normal map of the target model can be combined into a new map. When generating bas-reliefs, just the height of target model is constrained. Fig. 20 shows two bas-reliefs pasted on curved surfaces, where geometry details of the horses are well-preserved on a background (see the left image), and no height discontinuities appear on sharp edges (intersecting with the bunny) of the dodecahedron (see the right image).

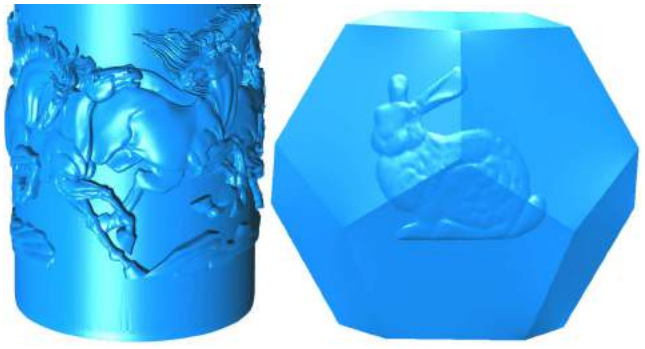


Fig. 20. Our bas-relief modeling on a cylinder background and a dodecahedron background, respectively. Geometry details of horses are well-preserved (left), and no gaps appear near intersections of sharp features (edges of the background) and the foreground (bunny) (right).

5.2 Detail Preservation and Enhancement

High compression of depth data will lose fine details of a 3D object. We use two normal maps to compensate for detail loss. The reason under our method is, 1) the first-order normal variations can better describe surface variations rather than vertex position variations, bas-relief modeling is thus taken place in normal image space; 2) details are preserved or enhanced by choosing a suitable value for μ in Eq. 6. Fig. 21 shows five models with multifarious details and their bas-relief modeling results (middle row). Details are all well-preserved though a large height compression is enforced. In addition, we can produce structure-preserving results (bottom row), if we just use the base normal map as input. More structure-preserving results can be found in Figs. 2 (the rightmost one), 6 (the rightmost two), 10 (the rightmost one) and 14 (the middle one).

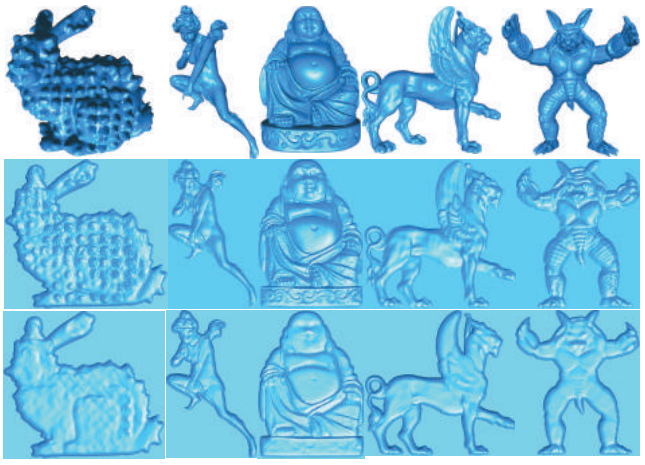


Fig. 21. Bas-relief modeling with the round style on the Bunny (middle: $\mu = 0.1, \beta = 0.01, \delta = 10$; bottom: $\mu = 0.1, \beta = 0, \delta = 15$), the Angel (middle: $0.1, 0.03, 10$; bottom: $0.1, 0, 15$), the Buddha (middle: $0.08, 0.06, 10$; bottom: $0.1, 0, 15$), the Armadillo (middle: $0.15, 0.05, 10$; bottom: $0.1, 0, 15$), and the Felion (middle: $0.12, 0.03, 10$; bottom: $0.1, 0, 15$) models.

We compare our method with the state-of-the-art methods, including Weyrich et al. [10], Sun et al. [9], Ji et al. [11], Schuller et al. [20], Zhang et al. [4], and all the results are rendered under the same lighting setting. Because most of them could not exactly control the depth of generated bas-reliefs, we use a linear scaling in the post-processing to make them have the same maximal height. The parameters for each method are carefully tuned to produce the best visual results under similar height compression. In Weyrich et al. [10], there are two parameters α and v_{sil} available, where α controls the degree of gradient compression, which is usually set to values between 0.5 and 10, and v_{sil} is the threshold of boundary gradient magnitude. In Sun et al. [9], there are four parameters used, where B is the number of bins of height values; m_0 is the minimum size of neighborhood, n is the number of neighborhood levels; α is the degree of the gradient compression. In Ji et al. [11], there are two parameters μ and θ, where μ is the compression coefficient, and θ is the height limit.

The results generated by the six methods share some similarities for many 3D scenes when the bas-reliefs with the standard thickness are produced, since details on these models are not largely lost, as shown in Fig. 22 and Fig 23.

There are many over-compressed bas-relief sculp-

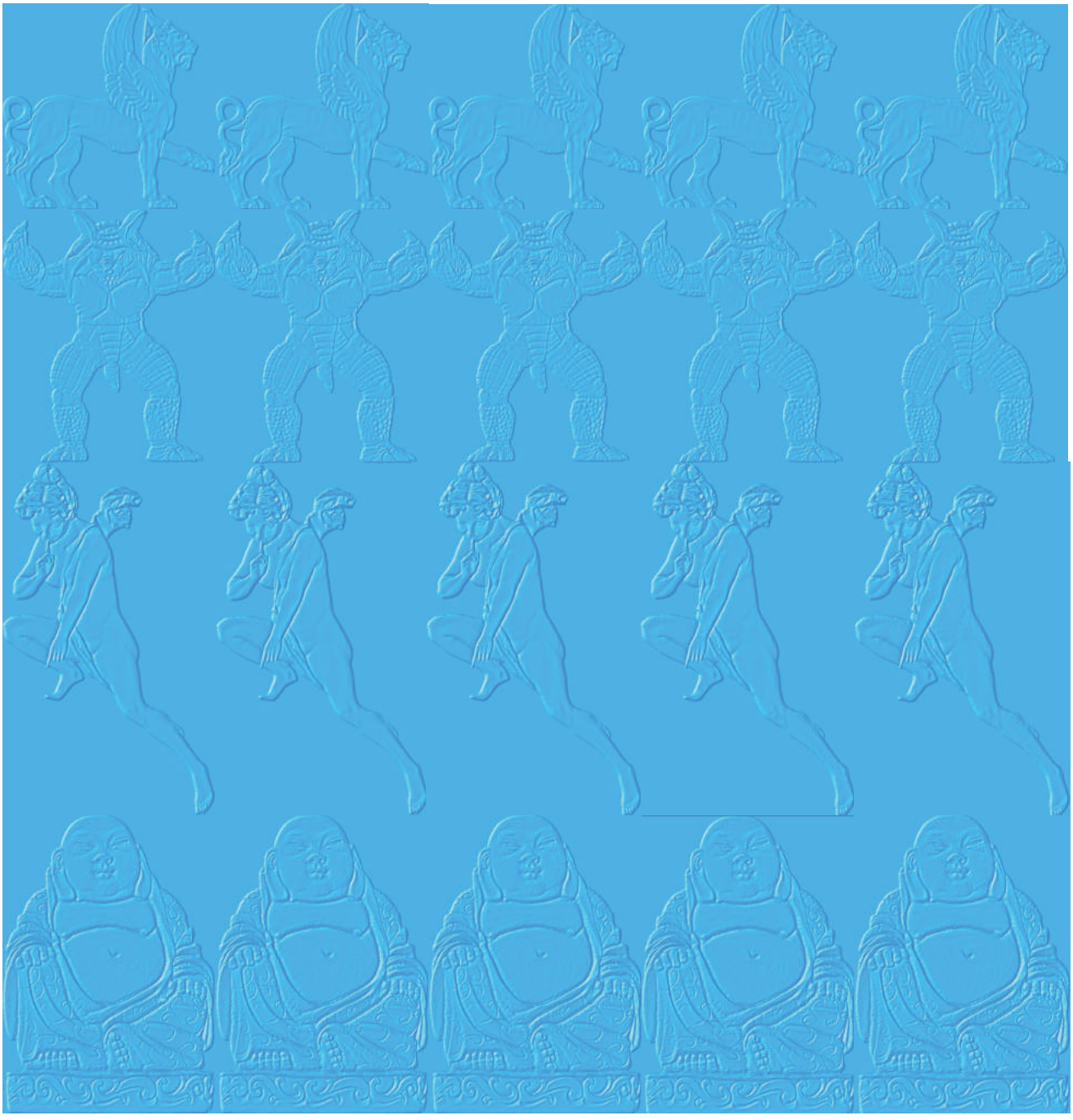


Fig. 22. Bas-relief modeling on four typical models with abundant details under a standard thickness. From the left column to the right: The results of Weyrich et al. [10], Sun et al. [9], Ji et al. [11], Schuller et al. [20], and our detail-enhancement results ($\mu = 0.1, \beta = 0.3, \delta = 10$). All results are with 10 high as the output height.

tures with vivid details in real life. For this special application, a slight advantage of our method is to better preserve the details, when the whole depth interval to be compressed is larger than the standard thickness. We now analyze the reason why our method outperforms the compared methods: The weight μ used in Eq. 6 can compensate the bas-reliefs for details lost, see Fig. 24, Fig. 25 and Fig. 26.

5.3 Computational Time

We record the time performance of each step in our experimental cases for some typical models. The results are shown in Tab. 1. It is observed that the preprocessing step is fast. However, we still suggest that this step is done in an off-line manner in practice: We can normal-filter a set of 3D surface meshes and save these models with each face assembled with a normal triple (N, N_B, N_D), where they represent

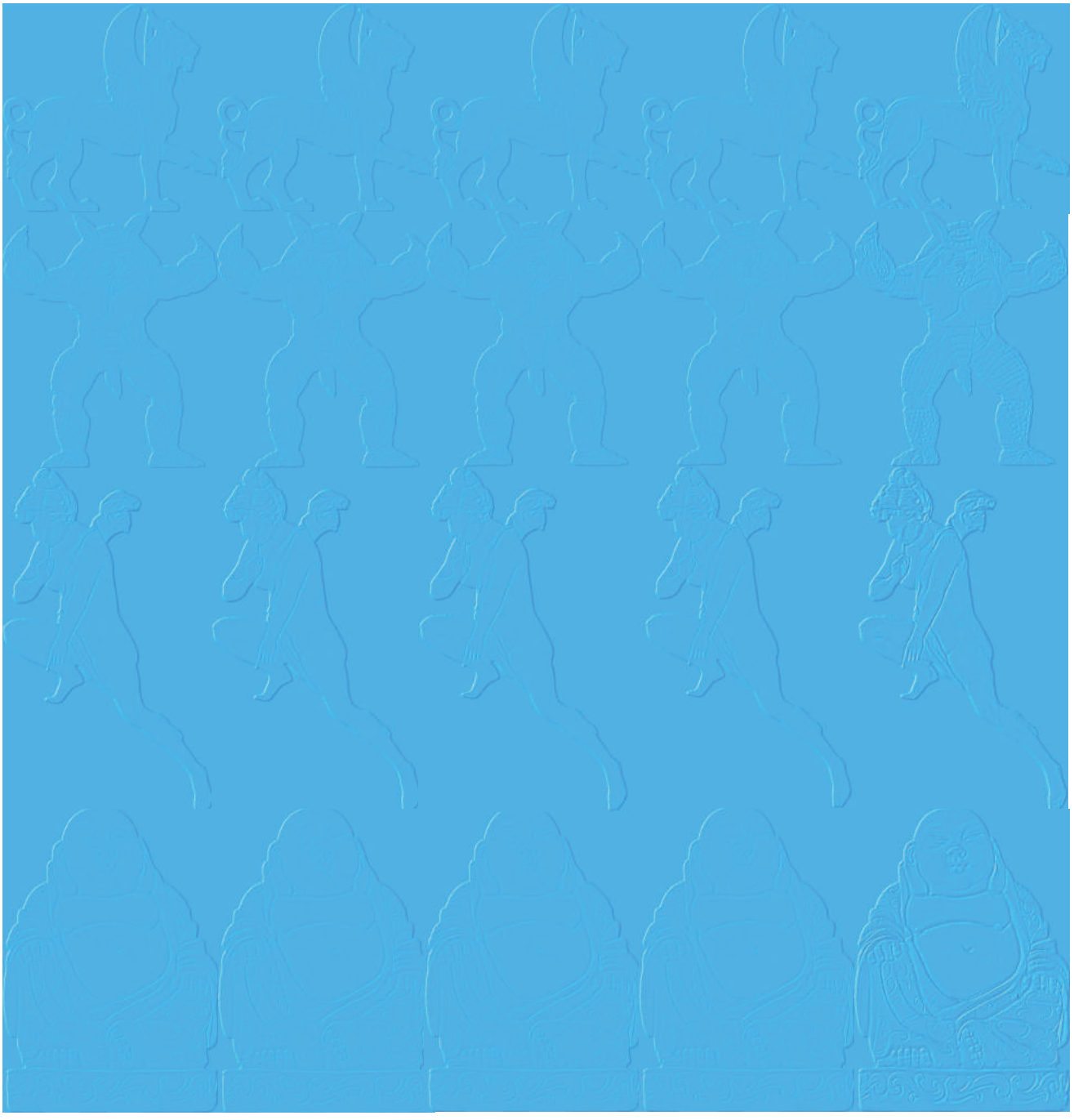


Fig. 24. Highly-compressed bas-relief modeling on four typical models with abundant details. From the left column to the right: The results of Weyrich et al. [10], Sun et al. [9], Ji et al. [11], Schuller et al. [20], and our detail-enhancement results ($\mu = 0.15, \beta = 1.2, \delta = 0.5$). All results are with 0.5 high as the output height.

the face's original normal, decomposed base normal and detail normal respectively. In bas-relief modeling, we need to solve a sparse linear system. We fixed the image resolution with 1000×900 for all tested models. Therefore, a sparse linear equations with size of around $3 \times 1000 \times 900$ should be solved. The factorization of matrix $A^T A$ can be reused for all models with the same image resolution, which saves the solution time. The second step, i.e., bas-relief modeling, is usually slower than the methods of Weyrich et al. [10] and Ji et al. [11], but faster than Sun et al. [9], Schuller et al. [20] and Zhang et al [4]. The time performance for our method is acceptable. For example, it takes a total of about 20.19s to model the Angel with 153k vertices and 307k facets and the image resolution of 1000×900. In the future, we will investigate how to accelerate the whole pipeline as presented in Zhang et al. [26] and Ji et al. [5].

Limitations. Our method has some limitations. First, we cannot exactly control the height range of

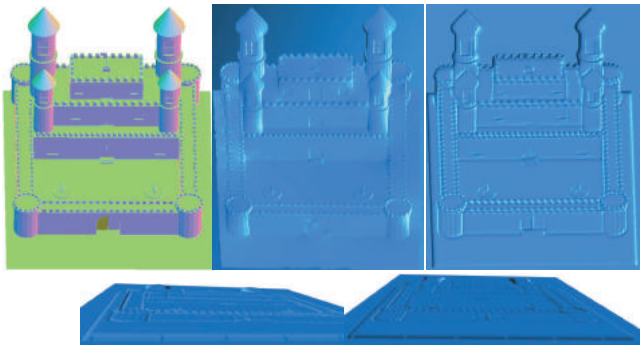


Fig. 23. Bas-relief modeling on castle. From the left column to the right: The input model, results of Zhang et al. [4], and ours ($\mu = 0.1, \beta = 0.3, \delta = 5$).

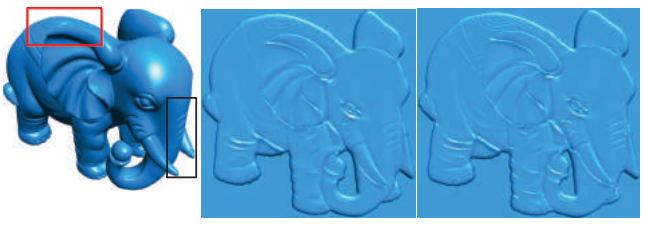


Fig. 25. Bas-relief modeling on elephant. From the left column to the right: The input model, the results Schuller et al. [20], and our detail-enhancement results ($\mu = 0.3, \beta = 0.2, \delta = 10$).

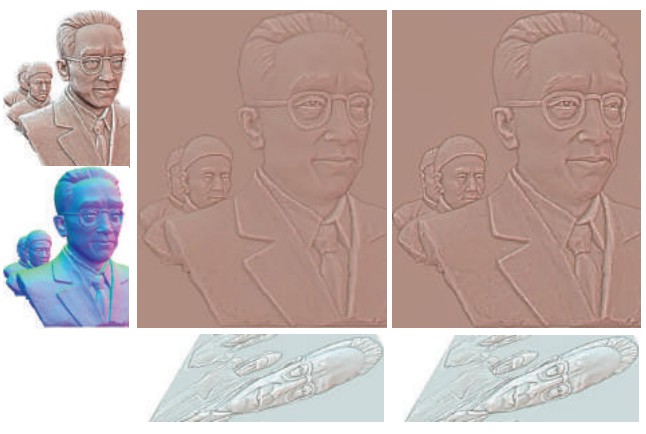


Fig. 26. Bas-relief set modeling. From the left column to the right: The first column shows the input's geometry and its normal map, the second shows the result of Ji et al. [11] and the third shows our result ($\mu = 0.2, \beta = 0.5, \delta = 2$).

TABLE 1
Timing of our method. In normal decomposition (stage 1), the numbers are the vertex and the facet numbers of each model; in bas-relief modeling (stage 2), the image resolution is fixed to 1000×900 for all models.

Models	Stage 1		Stage 2	Total
Angel	153k, 307k	1.34s	18.9s	20.19s
Armadillo	25k, 50k	0.39s	66.75s	67.14s
Buddha	757k, 1514k	6.8s	70.5s	77.3s
Felion	49.8k, 99.7k	0.56s	34.35s	34.91s
Elephant	171k, 342k	1.58s	67s	68.58s

our bas-reliefs. Interested readers can refer to Schuller et al. [20] that how to control the exact height for each vertex. Second, our method can input a depth image: A depth image can be regarded as a height field which can be compressed to a bas-relief. For our method, we convert the height field into a normal field (see [11] for conversion details), but we do not intend to detect normal discontinuities for the conversion. It will lead to over-sharpening of the high brightness regions in the depth image (see Fig.27), although their details are well-preserved. To solve this over-sharpening problem, we can detect the boundaries of these brightness regions. You can also select the well-known commercial software ArtCAM which processes a depth image. Third, the current version of our method does not support the composite modeling, i.e., high relief and bas-relief modeling for different parts of an input model. This may be achieved through the help of visual hulls [51].

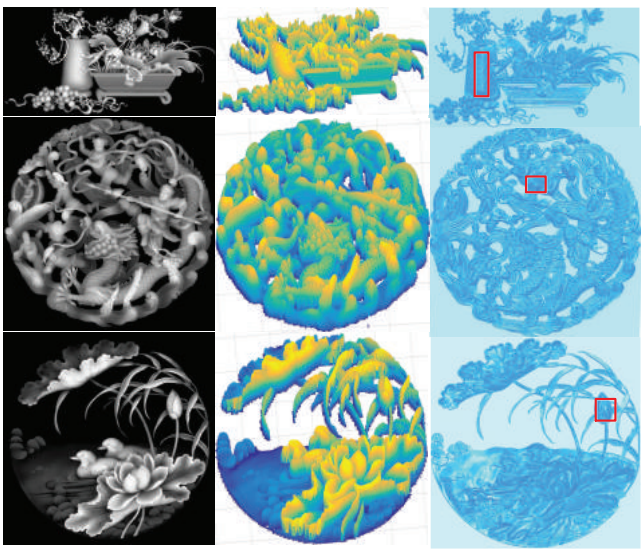


Fig. 27. Bas-relief modeling on Chinese style scenes taking 2 mm as the output height results. From left to right: the depth images, 3D illustration of these depth images (the yellow means the high brightness regions), the results of our method (the parts in red rectangles show the over-sharpening artifacts).

6 Conclusion

Digital bas-relief analysis is an important topic with the increasingly popular 3D scanning and printing techniques. This work aims to produce two kinds of bas-reliefs, one is detail-preserving bas-relief modeling, which can maintain a 3D model's overall shape and details. The other is structure-preserving bas-relief modeling, which just maintains the overall shape, while ignoring details. This is the main difference with most of the existing state-of-the art methods in this field.

It is well known that the underlying surface of an input mesh is commonly piecewise smooth and multi-scale geometry details exist within each smooth region. We decompose a surface in normal field into a base layer and a detail layer by the proposed GRNF. Existing filters (either isotropic filters or anisotropic filters) are usually enforced in a whole mesh model (face-by-face or vertex-by-vertex), a GMM analysis is required (this step is not intend to improve the existing filter, but to get two more accurate decomposed normal layers).

In addition, unlike existing bas-relief modeling methods, we not only transfer the continuous surface reconstruction into a discrete space, but also effectively solve it via a least-squares optimization step. The main contribution is a comprehensive way of building new solvers for two kinds of bas-relief modeling.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive suggestions. This work was supported by the grants from the National Basic Program of China, 973 Program (Project No. 2015CB351706), the National Natural Science Foundation of China (Project No. 61772267), the Research Grants Council of the HKSAR, China (Project Nos. 14225616 and UGC/FDS11/E03/14), and The Hong Kong Polytechnic University (Project No. 1-ZE8J).

REFERENCES

[1] D. Sýkora, L. Kavan, M. Cadík, O. Jamriska, A. Jacobson, B. Whited, M. Simmons, and O. Sorkine-Hornung, "Ink-and-ray: Bas-relief meshes for adding global illumination effects to hand-drawn characters," *ACM Trans. Graph.*, vol. 33, no. 2, pp. 16:1–16:15, 2014.

[2] F. Policarpo, M. M. Oliveira, and J. L. D. Comba, "Real-time relief mapping on arbitrary polygonal surfaces," *ACM Trans. Graph.*, vol. 24, no. 3, p. 935, 2005.

[3] A. A. Pasko, V. V. Savchenko, and A. Sourin, "Synthetic carving using implicit surface primitives," *Computer-Aided Design*, vol. 33, no. 5, pp. 379–388, 2001.

[4] Y. Zhang, Y. Zhou, X. Li, H. Liu, and L. Zhang, "Bas-relief generation and shape editing through gradient-based mesh deformation," *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 3, pp. 328–338, 2015.

[5] Z. Ji, X. Sun, S. Li, and Y. Wang, "Real-time bas-relief generation from depth-and-normal maps on GPU," *Comput. Graph. Forum*, vol. 33, no. 5, pp. 75–83, 2014.

[6] J. Kerber, M. Wang, J. Chang, J. J. Zhang, A. G. Belyaev, and H. Seidel, "Computer assisted relief generation-A survey," *Comput. Graph. Forum*, vol. 31, no. 8, pp. 2363–2377, 2012.

[7] R. Cromartie and S. M. Pizer, "Structure-sensitive adaptive contrast enhancement methods and their evaluation," *Image Vision Comput.*, vol. 11, no. 8, pp. 460–467, 1993.

[8] R. Fattal, D. Lischinski, and M. Werman, "Gradient domain high dynamic range compression," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 249–256, 2002.

[9] X. Sun, P. L. Rosin, R. R. Martin, and F. C. Langbein, "Bas-relief generation using adaptive histogram equalization," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 4, pp. 642–653, 2009.

[10] T. Weyrich, J. Deng, C. Barnes, S. Rusinkiewicz, and A. Finkelstein, "Digital bas-relief from 3d scenes," *ACM Trans. Graph.*, vol. 26, no. 3, p. 32, 2007.

[11] Z. Ji, W. Ma, and X. Sun, "Bas-relief modeling from normal images with intuitive styles," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 5, pp. 675–685, 2014.

[12] P. Wang, X. Fu, Y. Liu, X. Tong, S. Liu, and B. Guo, "Rolling guidance normal filter for geometric processing," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 173:1–173:9, 2015.

[13] J. Vollmer, R. Mencl, and H. Müller, "Improved laplacian smoothing of noisy surface meshes," *Comput. Graph. Forum*, vol. 18, no. 3, pp. 131–138, 1999.

[14] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa, "Laplacian mesh optimization," in *Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia 2006*, pp. 381–389, 2006.

[15] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," *ACM Trans. Graph.*, vol. 22, pp. 950–953, 2003.

[16] Y. Zheng, H. Fu, O. K.-C. Au, and C.-L. Tai, "Bilateral normal filtering for mesh denoising," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 10, pp. 1521–1530, 2011.

[17] X. Sun, P. L. Rosin, R. R. Martin, and F. C. Langbein, "Fast and effective feature-preserving mesh denoising," *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 5, pp. 925–938, 2007.

[18] S. Arpa, S. Süsstrunk, and R. D. Hersch, "High reliefs from 3d scenes," *Comput. Graph. Forum*, vol. 34, no. 2, pp. 253–263, 2015.

[19] X.-K. Yeh, S.-Y. Huang, P. K. Jayaraman, C.-W. Fu, and T.-Y. Lee, "Interactive high-relief reconstruction for organic and double-sided objects from a photo," *IEEE Trans. Vis. Comput. Graph.*

[20] D. Panozzo and O. Sorkine-Hornung, "Appearance-mimicking surfaces," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 216:1–216:10, 2014.

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

[22] W. Song, A. G. Belyaev, and H. Seidel, "Automatic generation of bas-reliefs from 3d shapes," in *2007 International Conference on Shape Modeling and Applications (SMI 2007)*, pp. 211–214, 2007.

[23] Y. Liu, Z. Ji, Z. Liu, and X. Wu, "Stylized design of bas-relief based on normal field," *Journal of Computer-Aided Design & Computer Graphics*, vol. 28, no. 12, pp. 2120–2127, 2016.

[24] Y. Zhang, C. Zhang, W. Wang, and Y. Chen, "Adaptive bas-relief generation from 3d object under illumination," *Comput. Graph. Forum*, vol. 35, no. 7, pp. 311–321, 2016.

[25] J. Kerber, A. Tevs, A. G. Belyaev, R. Zayer, and H. Seidel, "Real-time generation of digital bas-reliefs," *Computer-Aided Design and Applications*, vol. 7, no. 4, pp. 465–478, 2010.

[26] Y. Zhang, Y. Zhou, X. Zhao, and G. Yu, "Real-time bas-relief generation from a 3d mesh," *Graphical Models*, vol. 75, no. 1, pp. 2–9, 2013.

[27] Y.-W. Zhang, Y. Chen, H. Liu, Z. Ji, and C. Zhang, "Modeling chinese calligraphy reliefs from one image," *Computers & Graphics*, pp. 1–7, 2017.

[28] M. Alexa and W. Matusik, "Reliefs as images," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 60:1–60:7, 2010.

[29] Z. Li, S. Wang, J. Yu, and K. Ma, "Restoration of brick and stone relief from single rubbing images," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 2, pp. 177–187, 2012.

[30] L. Governi, M. Carfagni, R. Furferi, L. Puggelli, and Y. Volpe, "Digital bas-relief design: a novel shape from shading-based method," *Comput. Graph. Forum*, vol. 11, no. 2, pp. 153–164, 2014.

[31] J. Wu, R. R. Martin, P. L. Rosin, X. Sun, F. C. Langbein, Y. Lai, A. D. Marshall, and Y. Liu, "Making bas-reliefs from photographs of human faces," *Computer-Aided Design*, vol. 45, no. 3, pp. 671–682, 2013.

[32] J. Wu, R. R. Martin, P. L. Rosin, X. Sun, Y. Lai, Y. Liu, and C. Wallraven, "Use of non-photorealistic rendering and photometric stereo in making bas-reliefs from photographs," *Graphical Models*, vol. 76, no. 4, pp. 202–213, 2014.

[33] Q. Zeng, R. R. Martin, L. Wang, J. A. Quinn, Y. Sun, and C. Tu, "Region-based bas-relief generation from a single image," *Graphical Models*, vol. 76, no. 3, pp. 140–151, 2014.

[34] H. T. To and B. Sohn, "Bas-relief generation from face photograph based on facial feature enhancement," *Multimedia Tools Appl.*, vol. 76, no. 8, pp. 10407–10423, 2017.

[35] W. Xie, Y. Zhang, C. C. L. Wang, and R. C. Chung, "Surface-from-gradients: An approach based on discrete geometry processing," in *2014 IEEE Conference on Computer Vision and Pattern*

Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014, pp. 2203–2210, 2014.
[36] T. R. Jones, F. Durand, and M. Desbrun, "Non-iterative, feature-preserving mesh smoothing," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 943–949, 2003.
[37] L. Zhu, M. Wei, J. Yu, W. Wang, J. Qin, and P. Heng, "Coarse-to-fine normal filtering for feature-preserving mesh denoising based on isotropic subneighborhoods," *Comput. Graph. Forum*, vol. 32, no. 7, pp. 371–380, 2013.
[38] M. Wei, J. Yu, W. Pang, J. Wang, J. Qin, L. Liu, and P. Heng, "Bi-normal filtering for mesh denoising," *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 1, pp. 43–55, 2015.
[39] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, pp. 839–846, 1998.
[40] C. Bajaj and G. Xu, "Anisotropic diffusion on surfaces and functions on surfaces," *ACM Transactions on Graphics*, vol. 22, no. 1, pp. 4–32, 2003.
[41] U. Clarenz, U. Diewald, and M. Rumpf, "Anisotropic geometric diffusion in surface processing," in *IEEE Visualization*, pp. 397–405, 2000.
[42] A. E. Ouafdi and D. Ziou, "A global physical method for manifold smoothing," *Proceedings of Shape Modeling International*, pp. 11–17, 2008.
[43] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *Proceedings of Shape Modeling International*, vol. 12, no. 7, pp. 629–639, 1990.
[44] R. Wang, Z. Yang, L. Liu, J. Deng, and F. Chen, "Decoupling noise and features via weighted l1-analysis compressed sensing," *ACM Transactions on Graphics*, vol. 33, no. 2, p. 18, 2014.
[45] L. He and S. Schaefer, "Mesh denoising via l0 minimization," in *SIGGRAPH*, pp. 64:1–64:8, 2013.
[46] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via L_0 gradient minimization," *ACM Trans. Graph.*, vol. 30, no. 6, p. 174, 2011.
[47] Q. Zhang, X. Shen, L. Xu, and J. Jia, "Rolling guidance filter," in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III*, pp. 815–830, 2014.
[48] R. Zatzarinni, A. Tal, and A. Shamir, "Relief analysis and extraction," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 136:1–136:9, 2009.
[49] A. DEMPSTER, N. LAIRD, , and D. RUBIN, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.
[50] P. Wang, Y. Liu, and X. Tong, "Mesh denoising via cascaded normal regression," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 232:1–232:12, 2016.
[51] A. Laurentini, "The visual hull concept for silhouette-based image understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 2, pp. 150–162, 1994.

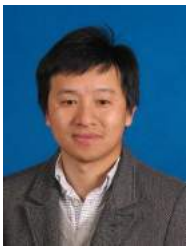
Mingqiang Wei received his Ph.D degree (2014) in Computer Science and Engineering from The Chinese University of Hong Kong (CUHK). He is an associate professor at Nanjing University of Aeronautics and Astronautics (NUAA)'s School of Computer Science and Technology. Before joining NUAA, he served as an assistant professor at Hefei University of Technology, and a postdoctoral fellow at CUHK. His research interest is computer graphics with an emphasis on smart geometry processing.

Yang Tian is now a PhD candidate at the Chinese University of Hong Kong's Department of Computer Science and Engineering. His research interests include computer graphics, and human-computer interaction.

Wai-Man Pang is an associate professor of the School of Computing and Information Science, the Caritas Institute of Higher Education, Hong Kong. He received his Ph.D. (2008) degree in Computer Science and Engineering from the Chinese University of Hong Kong. His research interests include non-photorealistic rendering, image & texture analysis, GPU computing, healthcare related simulations & applications and web-based Graphics.

Charlie C.L. Wang is currently a Professor and Chair of Advanced Manufacturing in the Department of Design Engineering at Delft University of Technology, The Netherlands. Prior to this position, he was a Professor of Mechanical and Automation Engineering at the Chinese University of Hong Kong (CUHK), where he started his academic career in 2003. Prof. Wang received awards from professional societies including the ASME CIE Excellence in Research Award (2016), the Best Paper Awards of ASME CIE Conferences (twice in 2008 and 2001 respectively), and the NAMRI/SME Outstanding Paper Award (2013). He serves on the editorial board of several journals, including Computer-Aided Design and IEEE Transactions on Automation Science and Engineering. His research interests are geometric computing, computer-aided design, advanced manufacturing, and computational physics.

Ming-Yong Pang is currently a professor at Nanjing Normal University. He received his Ph.D. (2004) degree in Computer-Aided Design from Jiangsu University, China. His research interests include computer graphics, geometry modeling and point cloud processing.



Jun Wang is currently a professor at Nanjing University of Aeronautics and Astronautics (NUAA), China. He received his Bachelor and PhD degrees in Computer-Aided Design from NUAA in 2002 and 2007 respectively. From 2008 to 2010, he conducted research as a postdoctoral scholar at the University of California and the University of Wisconsin. From 2010 to 2013, he worked as a senior research engineer at Leica Geosystems, USA. In 2013, he paid an academic visit to the Department of Mathematics at Harvard University. His research interests include geometry processing and geometric modeling.

Jing Qin received his PhD degree in Computer Science and Engineering from the Chinese University of Hong Kong in 2009. He has been an assistant professor at The Hong Kong Polytechnic University from 2016. His research interests include visualization, human-computer interaction and deep learning.

Pheng-Ann Heng received the PhD degree in computer science from Indiana University Bloomington in 1992. He is a professor at the Chinese University of Hong Kong's Department of Computer Science and Engineering. His research interests include VR applications in medicine, visualization, medical imaging, human-computer interaction, and computer graphics.