www.cadanda.com
Computer-Aided Design
AND APPLICATIONS

High Relief Modeling from 3D Object Through Laplacian-based Surface Deformation

Hao Zhou[1,2], Yanzhao Chen[1,2] , Ping Luo[1,2] and Yu-Wei Zhang[1,2]

[1]School of Mechanical and Automotive Engineering, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China
[2]Shandong Institute of Mechanical Design and Research, Jinan, China

Corresponding author: Yanzhao Chen, chyzh_ql@126.com

Abstract. High relief is a sculpture where more than half of 3D figure is attached onto a background plane. The main problem of high relief modeling from 3D object is how to transform the 3D geometry within limited depth range. This paper presents a novel method to generate high reliefs, which benefits from the technique of Laplacian-based mesh deformation. Given a 3D object as input, we first select a set of handle points on the input model and compute their offset distances to the background. Taking these handle points as constraints, we then optimize the depth field by solving a bi-Laplacian-based linear system. The deformed object is ensured to attach onto the background with preserved depth structure and geometrical details. Our method is effective in dealing with different types of input shapes, even the ones with topology-disconnected components. Experimental results and comparisons with previous method demonstrate the effectiveness of the proposed method.

Keywords: High Relief Modeling, Depth Compression, Mesh Deformation, Details Preservation
DOI: https://doi.org/10.14733/cadaps.2022.191-206

1 INTRODUCTION

High relief (Alto-rilievo) is a sculpture where more than half of 3D figure is attached onto a background plane as shown in Figure 1. In contrast to bas-reliefs, in which the scene elements are constructed in a very narrow depth range, high reliefs contain elements that are detached from the background. As show in Figure 1(a), some limbs are completely detached from the background, while the centaur's left rear leg is in the form of low relief. Compared with free standing 3D sculptures, high reliefs require less materials and are more easily fixed. Traditional creation of a high relief is laborious and relies much on the skills of artist. This paper aims to present a novel solution to facilitate high relief modeling from 3D object. The main challenge is how to select attaching points on the input object and preserve the 3D shape as well as geometrical details within limited depth range. As shown in Figure 2(b), straightforward compression-and-cutting-off removes important elements when viewed from the front. To preserve the original shape as much as possible, previous work [1] selects a set of attaching points and deform the object by moving the attaching points onto the background. However, in case the input object is composed of several disconnected parts, connectivity-aware surface deformation inevitably breaks the geometry, as show in Figure 2(c).

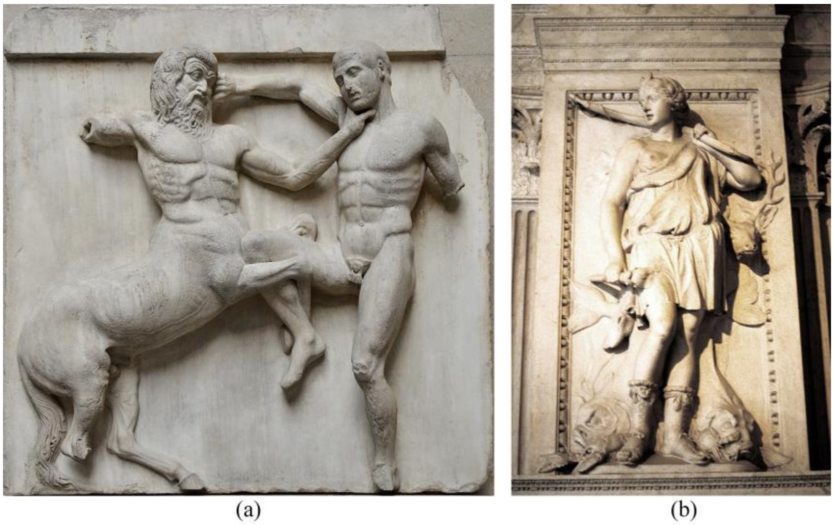

Figure 1: Artworks of high reliefs. (a) High relief metope from the Classical Greek Parthenon Marbles. (b) High relief from Donskoy monastery, Moscow, Russia.

To solve the above problems, we propose a two-stage solution in this paper. In the first stage, we search for two types of handle points through heuristic rules: back handles and connection handles. Back handles are located on the back of the model and defined as global constraints for surface deformation. We triangulate the back handles and compute their target positions to the background plane. In addition, connection handles are searched to stitch topology-disconnected parts together. Their target positions to the background are estimated with respect to the back handles. Taking target positions of these handle points as constraints, we finally optimize the 3D shape as well as geometrical details by solving a bi-Laplacian-based linear system. Figure 2(d) shows an example of the resulting high relief.

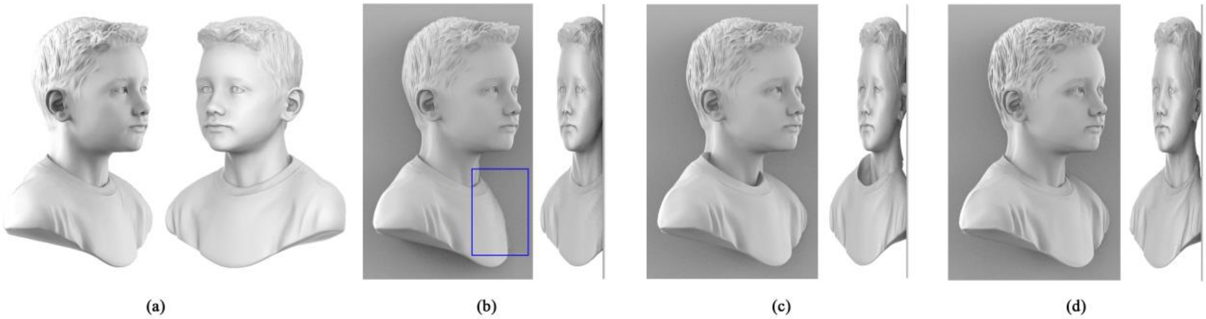

Figure 2: High relief modeling. (a) front view and side view of the input figure where head, eyeballs and body are visually joined but disconnected in topology. (b) geometrical details become blurred and right shoulder in the blue square gets lost in case the input object is simply compressed and cut off. (c) attaching handle points directly onto the background breaks the

geometry. (d) our method is effective in preserving geometrical details and depth structure of the input.

We validate the proposed method on a number of 3D models with varying shapes and topologies. Experimental results and comparisons with previous method show the effectiveness of our method. The rest of the paper is organized as follows. Related works are described in Section 2. Section 3 introduces technical details. In Section 4, we present experimental results and comparisons. Conclusions are given in Section 5.

2 RELATED WORKS

Reliefs are traditionally classified into bas-reliefs and high reliefs according to the depth that the figures are attached on the background. The definition is somewhat variable and many works combine more than one type of relief, sometimes sliding between them in a single scene. Bas-relief is usually represented as a 2.5D depth field with narrow depth range, a surface giving every point above the background a single height value. In contrast, a high relief has occluded features. The prominent elements in high reliefs are often undercut, detaching them from the background.

Traditional relief modeling using software tools such as Z-Brush and ArtCAM relies much on user interactions and skills. In the past few years, automatic or semi-automatic relief modeling from a 3D object or a 2D image has been a subject of interest in computer graphics. Among them, object-based methods take a 3D object as the input, and output the relief by transforming the object onto a background [2]. As the input object is fully rounded, users can freely change the viewing direction and create reliefs from desired ones.

For 2.5D bas-relief modeling, the main problems are how to eliminate the surface discontinuities in depth field and recover geometrical details due to depth compression. One category of existing solutions [3], [4], [5], [6] model bas-reliefs through gradient manipulation and Poisson-based surface reconstruction. Another category [7], [8], [9], [10], [11] approximates input surface normal while strictly obeying given depth constraints. Recently, [12] proposed a CNN-based method to generate bas-reliefs. As the network prediction is mainly based on convolutional operations, it runs much faster than traditional methods.

The above methods are not suitable for the modeling of high reliefs with undercut features. Instead, the main challenge for high relief modeling is how to select attaching points and preserve the 3D shape as well as geometrical details within limited depth range. In the pioneer work of [1], the authors select a number of attenuation points on front of the 3D object and move these points towards a relief plane by considering the local relationships. Taking new positions of attenuation points as constraints, they then reconstruct the surface through differential coordinates optimization. This method usually produces good results. However, it has some limitations. First, the overall depth range of high relief depends much on the number of attenuation points. It is prone to bring excessive surface distortions when the depth range is highly limited. Second, geometry details are still blurred although differential coordinates intrinsically encode geometrical details. Third, the input object is limited to unitary topology. Connection-aware surface deformation inevitably breaks the geometry in case the input object is composed of disconnected parts in topology. In this paper, we propose a different strategy to address these problems. Instead of selecting front attenuation points, we select handling points on the back of the input object. The deformation caused by the movements of handle points will be smoothly propagated to the front, thereby reducing distortions and improving visual quality of the front surface. Moreover, we add a normal-enhancement operation to preserve geometrical details in the final relief. For the topology-disconnected object, we construct a set of connection handles to join the neighboring parts, ensuring the depth relationships unchanged after surface deformation. Recently, [13] presented a framework for automatically creating sculpture paintings with high and bas-reliefs mixed within the same composition. As viewing angle changes, 3D elements can be seen emerging from the scene, producing interesting visual effect. Differently from this work, we assume that the figure in the high relief has no textures. Instead, we aim to attach the object onto the background while preserve the original appearance as much as possible.

Surface deformation is an important topic in geometry processing and has been studied for decades. Existing methods such as [14], [15], [16] rely on selecting a set of control points whose movements are interpolated to the rest mesh vertices. In this paper, we formulate high relief modeling as a problem of bi-Laplacian-based surface deformation. Compared to Laplacian-based deformation, the C1 continuity of bi-Laplacian operator avoids sharp cusps produced at handle points. Another possible solution for high relief modeling is to use cage-based deformation [17], [18], [19], where the input shape is enclosed by a coarse cage mesh and all surface points are written as linear combinations of the cage vertices. Cage-based deformation is robust in handling models with free topologies and effective in preserving geometric details. However, the storage of barycentric coordinates is quite memory-consuming.

3 METHODS

Our method benefits from bi-Laplacian-based surface deformation. The whole process consists of three main stages:

- Initial location (Section 3.1). Given a user-specified viewing direction, the input object is compressed and initially joined with the background.
- Handle points selection (Section 3.2 and 3.3). A set of back handles are selected from the back surface. In addition, connection handles are searched to join the neighboring parts in case the 3D model is composed of topology-disconnected elements. The target positions of all handle points are then estimated.
- Object attachment (Section 3.4). The object is deformed and attached onto the background by solving a bi-Laplacian-based linear system with target constraints.

3.1 Initial Location

Before surface deformation, we first need to locate the object so that it is close enough to the background while the desirable elements are still visible when viewed from front. Given a user-specified viewing direction, we rotate the 3D object around its center and make sure the viewing direction matches with negative z-axis in world coordinate system. Then, we detect the visibilities of mesh vertices via depth sampling, and classify the vertices into two categories: Back Vertices and Front Vertices. The Back Vertices are the ones which are not occluded by any elements when viewed from back (blue in Figure 3). The rest ones are denoted as Front Vertices. We assume the background plane is located at coordinate center with zero depth. Among all the front vertices, the vertex with minimum depth coordinate will be selected as reference point. To ensure all front vertices still stand in front of the background, we translate the object in depth direction and align the reference point with the background plane. After that, we linearly compress the depth coordinate by:

$$d_i = \alpha \cdot \overline{d_i} \tag{3.1}$$

Where α is the compression ratio and $\overline{d_i}$ is the translated depth value. Instead, non-linear function [1] can also applied to compress the front vertices and back vertices with different degrees. However, we find that the final high relief has little difference with the one using linear compression, especially for the case with large compression ratio. Figure 3 shows our location result, where the object has been initially joined with the background. However, most back elements are still detached. For reliable installation, we will select more back vertices and attach them onto the background. At the same time, we will recover the lost geometrical details due to depth compression.

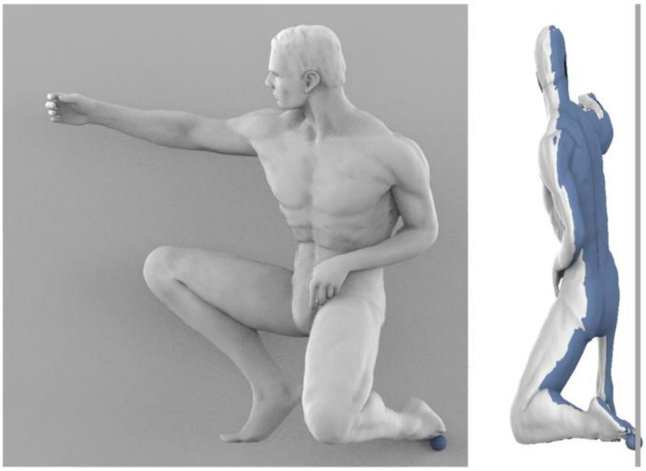

Figure 3: Initial location. The Front Vertex with minimum depth coordinate is selected as a reference point (blue ball). Then, the 3D object is transformed by aligning the reference point with the background plane. Back Vertices are colored in blue.

3.2 Back Handles Selection

In this section, we select a set of handle points and compute their target positions to the background. As our target is to bring more back elements to the background, we leave out the front vertices and only select handle points on the back surface. At the beginning, we propose the following heuristic criteria:

- The normal of a back handle should be as nearly perpendicular to the background as possible.
- A back handle should be locally convex in 3D space.
- A back handle should keep a distance with the vertices behind the background plane.

To meet the above criteria, we compute the angle of normal vector with negative z-axis and the mean curvature for each back vertex. When the vertex has positive mean curvature and the normal angle is smaller than a predefined threshold ϑ , we further compute its minimal Euclidean distance from the vertices behind the background. The ones whose distances are larger than a predefined threshold v_d will be selected as candidates, as shown in Figure 4(a). However, the candidate points are still too dense. To reduce the number, we additionally create $N \times N \times N$ uniform voxel grids. For a non-empty voxel that contains a number of candidates, we choose the one closest to the voxel center as a Back Handle, as shown the black dot in Figure 4(b). In our experiments, we set ϑ = 10°, v_d =0.1 and N =20, which works well in our experiments.

Self-intersections and excessive distortions will be created if all back handles are attached onto the background through surface deformation. To avoid this, we only select the points closer to the background plane as attaching points. For easy adjustment, we normalize the depths of back handles within the range of [0.0, 1.0], and choose the ones below a distance threshold t , as shown the blue dots in Figure 4(c). Here, not all attaching points are necessary for the purpose of installation. Instead, we wish the attaching points are more likely to appear on the top and bottom, which is stabilized enough to fix the object. To achieve this, we additionally partition the points into m (m =10 by default) uniform groups in x-direction. In each group, the attaching points with maximum and minimum y- coordinates are kept while the middle ones are discarded, as shown in Figure 4(d).

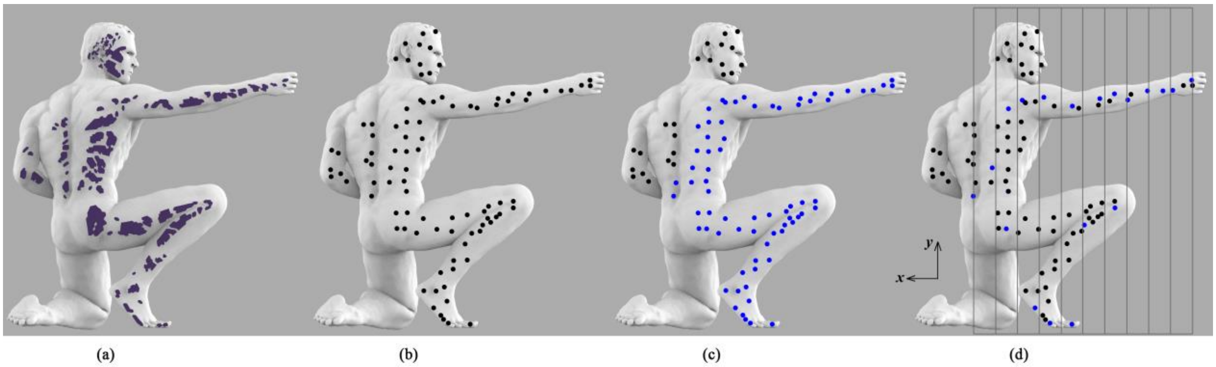

Figure 4: Progressive back handles selection. (a) dense back vertices are first selected by heuristic criteria. (b) sparse back handles are then selected through voxel-based clustering. (c) attaching points (blue dots) are selected from the back handles with a distance threshold t. (d) the highest and lowest dot in blue are selected as the final attaching points in 2D grids while the middle ones are discarded.

Next, we estimate target positions of the back handles, which will be used afterwards as depth constraints for surface deformation. To achieve this, we first apply 2D Delaunay triangulation to link the back-handle points. As the destinations of the attaching points are known (i.e., zero depth), we take them as boundary conditions and estimate the movements of other handle points by solving the following harmonic equation:

$$L \cdot \delta d = 0 \tag{3.2}$$

where L is weighted Laplacian matrix. After solving Equation (3.2), the target depth coordinates are computed by adding the movements to the original ones.

Taking all back handles as constraints strengthens the rigidity of deformation. To verify this, we do an alternative experiment that only takes the attaching points (blue dots in Figure 4(d)) as depth constraints for surface deformation. Figure 5 shows the resulting high relief. Unfortunately, the right hand has been wrongly moved inside the body. In contrast, the result with full back handles preserves the original depth structure as rigid as possible.

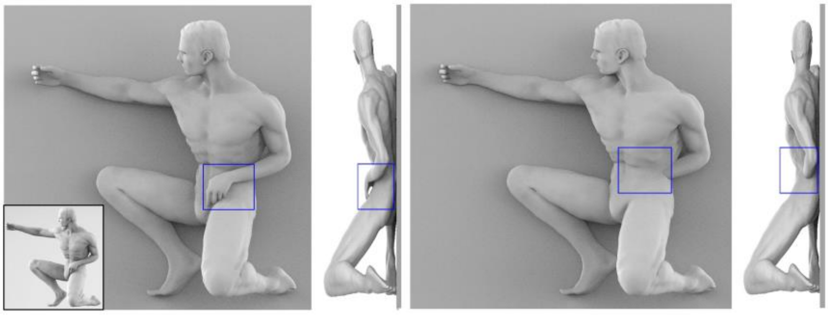

Figure 5: Deformations with different depth constraints. Right: result by applying all back handles as depth constraints, where the depth structure remains unchanged. Left: result by only applying attaching points as depth constraints, where the right-side hand has been wrongly moved inside the body. Inset image shows the input object.

3.3 Connection Handles Construction

It is sufficient to apply mesh deformation with given handle constraints so that a number of back elements are attached onto the background. However, it is common for hand-sculpted models that different parts of a 3D object are visually joined but topological disconnected. For these objects, connectivity-aware surface deformation inevitably breaks the geometry. One possible way to solve this problem is to remesh the 3D object in a preprocessing stage and convert it into union 3D form, but the additional operation will be very time-consuming. In this paper, we wish to directly apply deformation on the original object. Therefore, we need to consider geometrical relationships between neighboring parts. Usually, it is easy to make a rule that two vertices belong to different parts are geometrical related if their Euclidean distance is below a threshold. However, constructing such relationships vertex-by-vertex is quite inefficient because the computational complexity equals to $o\left(n^2\right)$, where n is the number of mesh vertices.

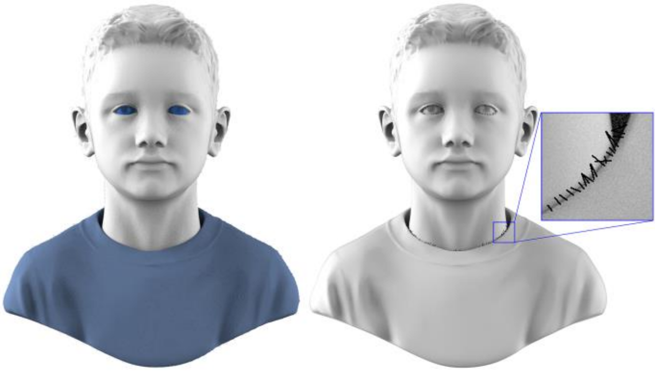

Figure 6: Connection handles. Left: 3D object with head, body and eyes visually joined but disconnected in topology (visualized in different colors). Right: stitching different parts together by a set of connection handles.

In our implementation, we first search the part index for each vertex using flood-filling algorithm. Starting from a random point and a new part index, we iteratively flood the current point to its one-ring neighbors until all vertices have been looped over. Thus, each vertex will have a unique part ID. Then, we again create $N \times N \times N$ uniform voxel grids in a unit cube. We loop every non-empty voxel and check whether its bounding vertices has different parts IDs. If it is true, we compute the minimal Euclidean distance for each vertex with its neighbors that have different part IDs in this voxel. Two vertices will be virtually joined if their distance is below a given threshold ε , as shown in Figure 6. In this way, the computational complexity is largely reduced. By default, we set ε as the average edge length and N =20.

We name these points as Connection Handle and use them to enforce additional constraints for surface deformation. This guarantees neighboring parts stitched without a requirement of unitary topology. Once the connection handles have been searched, we compute their depth movements to the back ground by a weighted average on the movements of the back handles:

$$\delta d_i = \sum_{j=1}^{m} w_{ij} \cdot \delta d_i \tag{3.3}$$

where δd_j is the depth movements of the back handle, w_{ij} is the vertex-wise weight that is defined by the inverse Euclidean distance from connection handle v_i to the back handle v_j , and the sum of weight is regulated by $\sum_{j=1}^{m} w_{ij} = 1.0$.

3.4 Object Attachment

Similar to the work of [20], we formulate high relief modeling as a problem of Laplacian-based surface deformation, which is expected that the depth structure can be preserved as much as possible while the lost surface details due to depth compression can be recovered.

We enhance normal details of the input object to compensate the loss of geometrical details. We first smooth the surface normal by Laplacian filtering. For each vertex whose normal facing to the front, we then scale up the angle between the original normal n_i and the filtered normal n_i^o :

$$n_i' = Rot\left(n_i^o, \beta \cdot \left\langle n_i, n_i^o \right\rangle\right) \tag{3.4}$$

where *Rot* is the rotation operator, $\left\langle n_i, n_i^o \right\rangle$ is the normal angle and β is the scaling factor to control the intensity of normal detail. After that, we compute the target normal of a triangle $n(t_i)$ by the sum of three vertex normals and normalization, which will be used to compute the depth divergence in Equation (3.6).

Bring elements onto the background can be regarded as the following optimization problem:

$$d' = argmin\sum_i \left(\nabla d_i' - g_i\right)^2 \tag{3.5}$$

where g_i is the target gradient, $\Delta d_i'$ is the gradient of depth coordinate and d_i' is the depth coordinate needs to be optimized. Minimizing the energy function equals to solve a sparse linear system:

$$L \cdot d' = div(g) \tag{3.6}$$

where L is the 1-ring Laplacian matrix and $div(g)$ is the target divergence vector. We further modify Equation (3.6) by multiplying L^T on both sides of the Equation, thereby achieving a bi-Laplacian linear system:

$$L^2 \cdot d' = b \tag{3.7}$$

where L^2 is the 2-ring bi-Laplacian matrix and b is the bi-divergence. Please refer to [14] for the definitions of Laplacian matrix and bi-Laplacian matrix. As L^2 is known, we need to estimate b with respect the enhanced surface normals. In details, we virtually rotate each triangle around its center and make its normal to be aligned with the target normal $n(t_i)$. The 1-ring depth divergence $div(g)$ for each vertex is calculated according to the method of [14]. At last, we compute the bi-divergence vector b by multiplying Laplacian matrix L^T with $div(g)$. Note Equation (3.6) and Equation (3.7) are equivalent, but the resulting surface around the handle points varies a lot. As shown in Figure 7(b), cusp points will be produced by solving Equation (3.6). In contrast, the C1 continuity of bi-Laplacian operator in Equation (3.7) ensures the deformation smoothly propagated at the handle points, thereby avoiding cusps, as shown in Figure 7(c).

As mentioned above, parameter β is used to control the intensity of geometrical details in the final relief. Increasing β brings more details recovered. However, an overlarge value might break the original appearance, as shown in Figure 8(d). In our implementation, we suggest β =1.0 when the compression ratio α is larger than 0.6. Otherwise, we set β =2.0 by default.

4 RESULTS

We implemented the proposed method using C++ and OpenGL, and tested it on a PC with an Intel(R) Core(TM) i5-3230M CPU@2.6 GHz and 2.88 GB of RAM. In our modeling tool, users can freely choose the viewing direction by rotating the input 3D object. Once a desired direction is determined, our method computes target positions of the handle points and attaches the input model onto the back ground based on the default parameters.

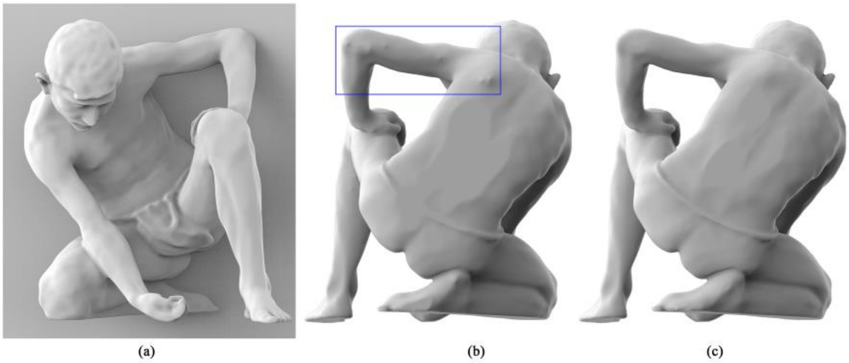

Figure 7: Comparison of Laplacian-based surface deformation and bi-Laplacian-based surface deformation. (a) front view of Laplacian-based deformation. (b) result of Laplacian deformation when viewed from back, where cusps are produced at the handle points. (c) the C1 continuity of bi-Laplacian deformation ensures C1 continuity at handle points, thereby avoiding cusps.

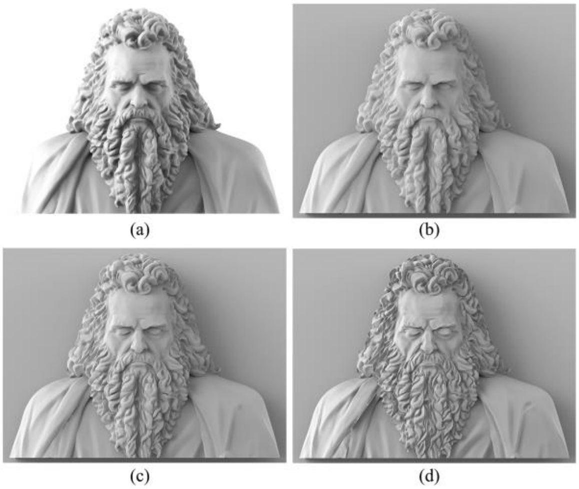

Figure 8: Details enhancement with different scaling factors. (a) input object. (b) β =1.0. (c) β =2.0. (d) β =4.0. All reliefs are generated with same compression ratio α =0.4. Note that the details are not recovered in (b) while the details in (d) are extremely exaggerated. We suggest β =2.0 when the compression ratio α is smaller than 0.6.

For an input object with 20k vertices, the whole computation can be completed in 15 seconds, which is acceptable for real applications. To make the modeling more flexible, we allow users to manually adjust the compression ratioα and distance threshold t , add or delete handle points before surface deformation. Demo video is available at:
https://drive.google.com/file/d/1Fu9xuq8WZ8I1Nla4ZQvc1c1z5DRxMFMn/view

4.1 Depth Range Control

Compression ratio α in Equation (3.1) is used to control the overall depth range of the high relief. In Figure 9, we show several results from same object and viewing direction, but with different settings of α . Note that the two wings and the body in Figure 9(a) have different distances from the background while all of them have been attached onto the background in the relief. As expected, all results show similar appearance to the input when viewed from front. To quantitively evaluate the difference, we compute mNA (mean normal angle) between the vertices of the attached object and the input object. The mNAs in Figure 9 are 10, 15, 25 degrees respectively, which rise with the decrease of α .

Note that we first compress depth range of the input object and then optimize the depth coordinates. The resulting high relief is different from the one by deformation-before-compression. The former is more efficient in recovering surface details, but the expense is that the depth range cannot be accurately controlled. In most cases, the depth range is slightly degenerated due to the movement of scene elements. It is possible to further compensate the loss with a scaling-up operation, but the surface details might be overly exaggerated. We omit this problem in our implementation.

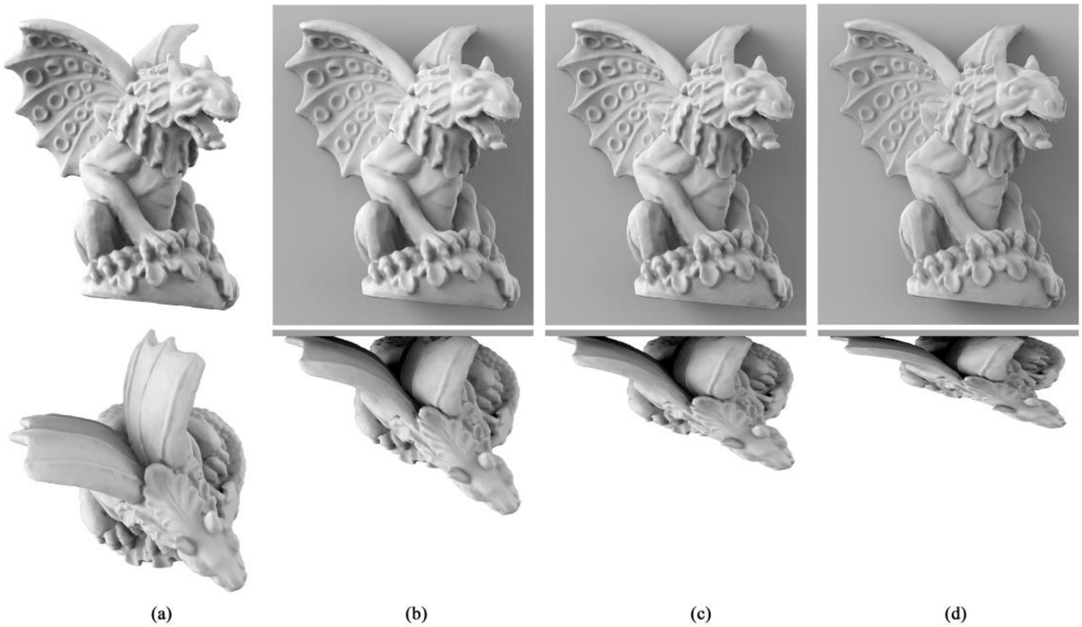

Figure 9: Results with different compression ratios (a) input object (b) α =0.8 (c) α =0.6 (d) α =0.4.

4.2 Results with Different Viewing Directions

Viewpoint selection for 3D object is a classic problem in computer vision [21]. For high relief modeling, this problem is more challenging because many factors such as pose, geometry, saliency and installation should be involved, which is beyond the scope of this paper. In our modeling system, we require users to specify desired ones according their own intentions. In Figure 10, we show several results with different viewing directions from the same input. In each result, a number of surface points has been successfully attached onto the background while the depth structure and geometrical details are well preserved.

4.3 Impact of Attaching Points

Distance threshold t in section 3.2 is used to control the positions of attaching points. Increasing t will bring more points attached on the background.

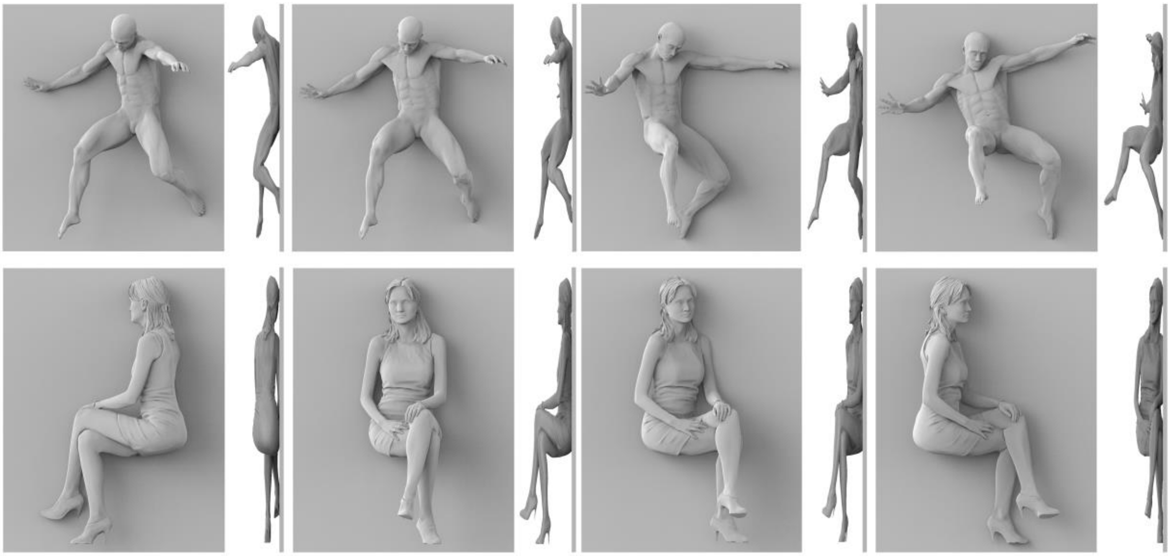

Figure 10: Results with different viewing directions. Compression ratio ɑ=0.4.

In our modeling system, we provide users a slide tool to tune parameter t and update the positions of attaching points. Users can also add or delete handing points on the back surface. Figure 11 show several results with t =0.2, t =0.7 and t =1.0 respectively. In each result, we visualize the normal angles between the attached object and the initial object (Figure 11(a)) by a color map, Usually, large normal variations (red color) are produced on the elements with large movements.

4.4 Varying Shapes and Topologies

Our method is capable in handling different categories of 3D shapes, as the results shown in Figure 12. The initial objects are obviously not faithful for installation because only few points are joined with the background. After modeling, more solid junctions have been constructed between the objects and the backgrounds in the high reliefs. In Figure 13, we show two examples generated from topology-disconnected shapes. Specially, the skeleton horse has 200 individual bones. Despite of complex topology, the relative positions of the bones are well preserved after surface deformation. To verify manufacturability of the results, we fabricated several models by 3D printing using photosensitive resins as shown in Figure 14.

4.5 Comparisons and Discussions

To make comprehensive comparisons with the method of [1] who also model high reliefs from 3D objects, we generate two reliefs with different depth ranges. The first one is generated with compression ratio α =0.7 and the other one is limited with α =0.35, as shown in Figure 15(c) and Figure 15(e). For fair comparisons, both of them have equivalent depth ranges with the results in [1]. Since our target is to make the relief looking as close as possible to the input object, we evaluate the similarity by using Mean Normal Angle (mNA) and PSNR of the normal maps. Larger PSNR and smaller mNA indicate greater similarity between the relief and the input object. As shown in Table 1, the mNA and PSNR of our result in Figure 15(c) have small differences with the result of [1] in Figure 15(b). However, comparing the small-depth-range examples in Figure 15(d) and Figure 15(e), our result performs better both in mNA and PSNR.

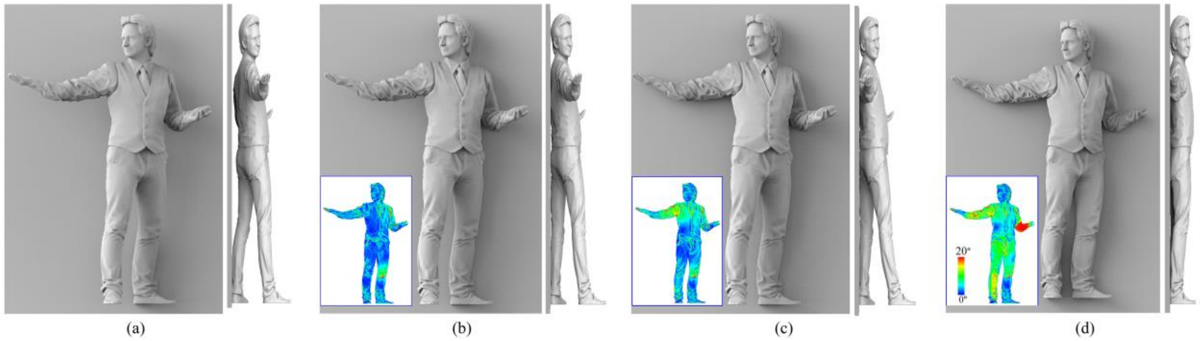

Figure 11: High reliefs with different settings of distance threshold t. (a) initial object (b) t=0.2 (c) t=0.7 (d) t=1.0. The color map shows the normal angle variations between the vertices of the attached model and the initial model in (a).

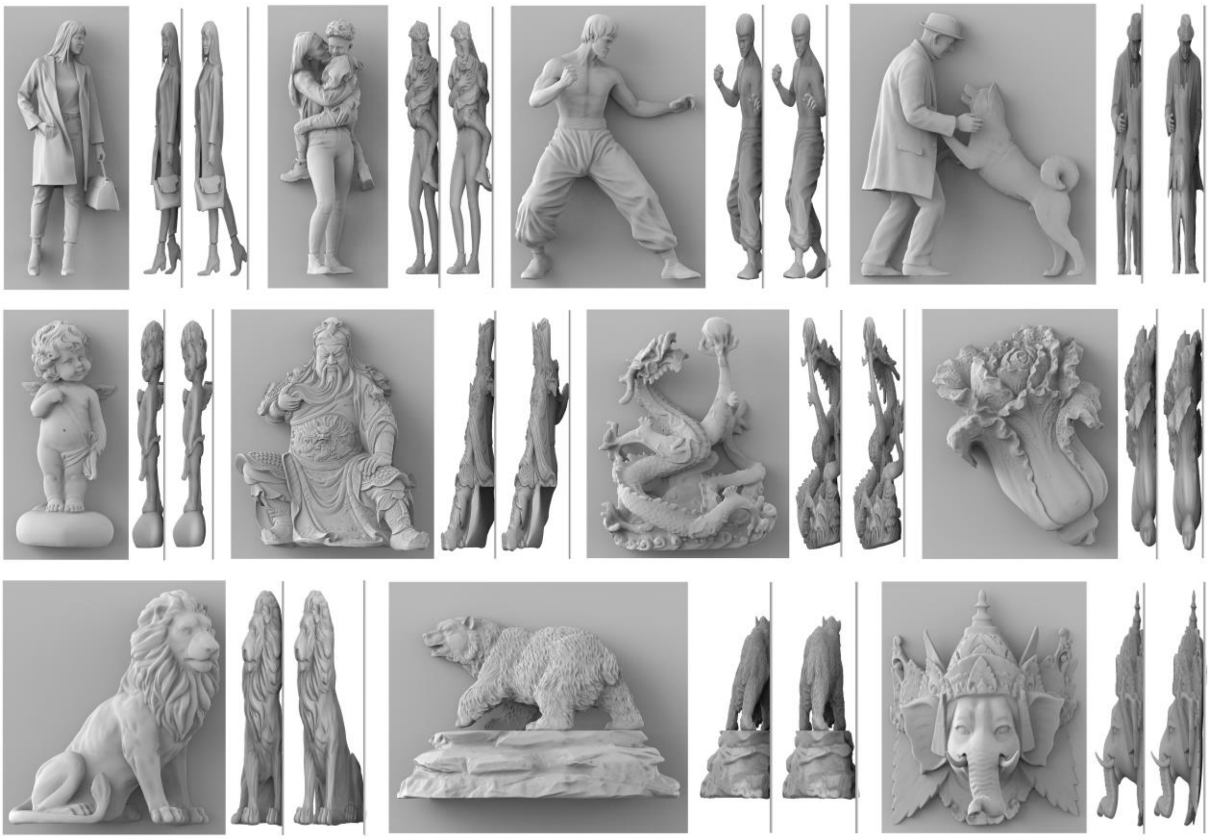

Figure 12: High reliefs generated from different 3D shapes. From left to right in each example: font view of the attached object, side view of the attached object and side view of the initial object before surface deformation.

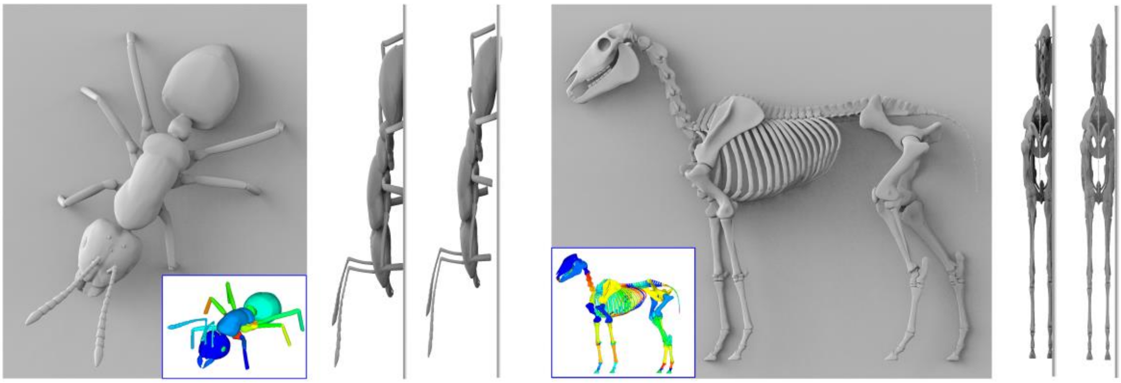

Figure 13: Results generated from topology-disconnected shapes. Each example contains a number of individual parts (visualized in different colors), which are joined together in the final relief. The rightmost views show the initial objects before surface deformation.

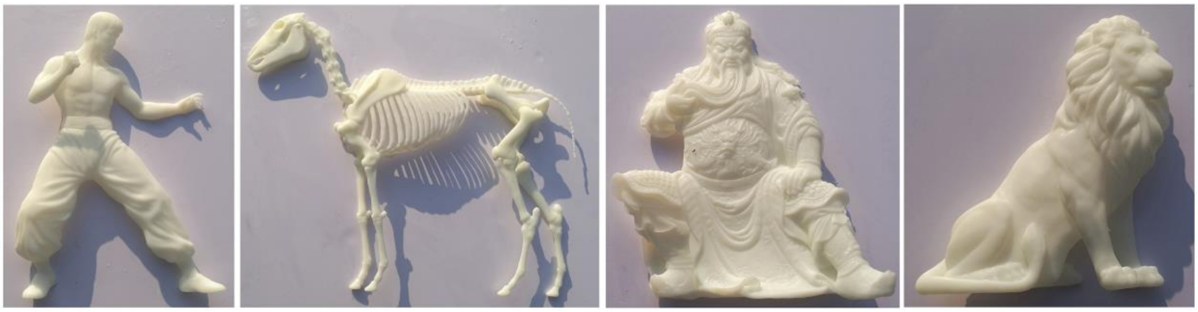

Figure 14: High reliefs fabricated by 3D printing.

Visual results match the above quantitative evaluations. The result of [1] in Figure 15(d) has distinct surface distortions, particularly at the belly region. Comparably, our result in Figure 15(e) looks more satisfactory with enhanced surface details.

Now, we discuss the difference between Figure 15(d) and Figure 15(e). In the work of [1], the authors first select attenuation points on the front surface. Then, they nonlinearly compress the distances of the attenuation points. Lastly, they deform the whole model by bringing the attaching points onto the background. To further limit the depth range, a nonlinear function is additionally applied to compress the depth field. However, applying deformation and compression directly on the front surface brings perceptible surface distortions, as shown the belly region in Figure 15(d). In contrast, we select handle points on the back surface for surface deformation. We first compress the depth range of the input model, and then optimize the depth range by moving the attaching points onto the background. The energy caused by handle movement will be smoothly propagated to the front surface. Thus, even when the depth range is highly compressed, the whole depth structure can still be well preserved.

Details preservation is another important issue for high relief modeling. Both methods generate high reliefs by applying Laplacian-based surface deformation, which intrinsically encodes geometrical details of the input. However, the intensity of surface details in Figure 15(d) is very limited. We solve this problem by adding a normal-enhancement operation, thereby strengthening surface details in the final relief, as shown in Figure 15(e).

Different with the method of [1] that can only deal with 3D object with unitary topology, our method is also capable in handling topology-disconnected objects. Thus, a wider range of 3D models can be directly utilized for high relief modeling. Our method runs much faster because we reconstruct the surface only once. In contrast, the method of [1] applies additional reconstruction to remove local cusps. For the object with 23k vertices in Figure 15, their method takes about 54 seconds to generate the high relief, while our method only takes only 15 seconds.

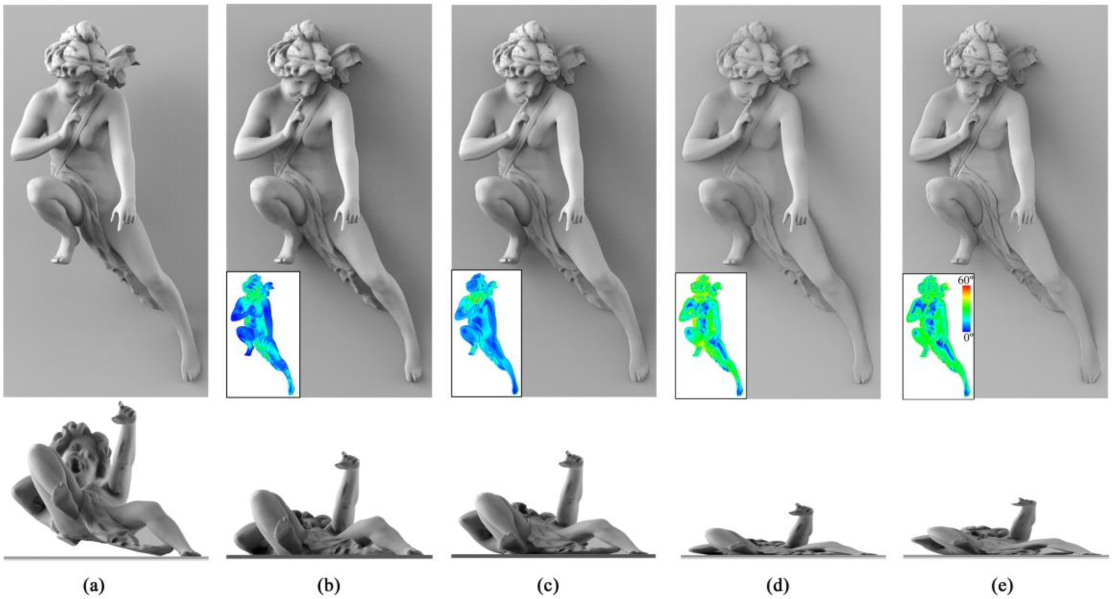

Figure 15: Comparisons with previous method. (a) input object. (b) result of [1] with no depth compression, attenuation points number: 128. (c) our result with depth compression α =0.7 and distance threshold t =0.7. (d) result of [1] with compression ratio 0.5, attenuation points number: 128. (e) our result with depth compression α =0.35 and distance threshold t =0.7. The inset images represent the normal angle variations between the reliefs and the input object.

	The result of [1] in Figure 15(b)	*Our result in Figure 15(c)*	*The result of [1] in Figure 15(d)*	*Our result in Figure 15(e)*
Mean Normal Angle (◦)	11.35	**10.94**	25.10	**22.29**
PSNR	29.65	**30.00**	25.32	**26.04**

Table 1: Statistics of comparisons.

5 CONCLUSIONS AND FUTURE WORKS

In this paper, we present a novel method to model high reliefs from 3D objects. Given a user-specified viewing direction, we first search a set of handle points on the back surface of the input object and compute their target positions to the background. In case the object is composed of topology-disconnected parts, we further define connection handles to link the parts together. Taking the handle points as depth constraints, we finally deform the object by solving a bi-Laplacian linear system. The optimized object is ensured to attach onto the background with preserved shape and geometrical details.

Although our method achieves better result than previous method, it still has some limitations. First, the surface attachment depends on variational mesh deformation. It is not suitable to CAD-like objects which have sharp edges and unregular mesh triangles. An alternative solution is to apply cage-based deformation, but the memory-consuming problem needs to be solved. Second, we propose heuristic rules to select attaching points which lacks of theoretical justification. Although the resulting high relief is visually satisfactory, it might not be the optimal one for stable installation. In the future, we will continue to explore this challenging problem.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their reviews and valuable suggestions. This work was supported in part by the National Natural Science Foundation of China (Grant No. 61772293).

Yanzhao Chen, https://orcid.org/0000-0002-5657-413X
Yu-Wei Zhang, https://orcid.org/0000-0001-6566-5714

REFERENCES

[1] Arpa, S.; Susstrunk, S.; Hersch, R.D.: High reliefs from 3d scenes, Computer Graphics Forum, 34(2),2015, 253–263. https://doi.org/10.1111/cgf.12557

[2] Zhang, Y.W.; Wu, J.; Ji, Z.; Wei, M.; Zhang, C.: Computer-assisted relief modelling: A comprehensive survey, Computer Graphics Forum, 38(2), 2019, 521–534. https://doi.org/10.1111/cgf.13655

[3] Weyrich, T.; Deng, J.; Barnes, C.; Rusinkiewicz, S.; Finkelstein, A.: Digital bas-relief from 3d scenes, ACM Transactions on Graphics, 26(3), 2007, 32–38. https://doi.org/10.1145/1275808.1276417

[4] Song, W.; Belyaev, A.; Seidel, H.P.: Automatic generation of bas-reliefs from 3d shapes, In: IEEE International Conference on Shape Modeling and Applications, 2007, SMI ’07, 2007, https://doi.org/10.1109/SMI.2007.9

[5] Bian, Z.; Hu, S.M.: Preserving detailed features in digital bas-relief making, Computer Aided Geometric Design, 28(4), 2011, 245–256. https://doi.org/10.1016/j.cagd.2011.03.003

[6] Zhang, Y.W.; Zhou, Y.; Li, X.; Liu, H.; Zhang, L.: Bas-relief generation and shape editing through gradient-based mesh deformation, IEEE Transactions on Visualization and Computer Graphics, 21(3), 2015, 328–338. https://doi.org/10.1109/TVCG.2014.2377773

[7] Ji, Z.; Ma, W.; Sun, X.: Bas-relief modeling from normal images with intuitive styles, IEEE Transactions on Visualization and Computer Graphics, 20(5), 2013, 675–685. https://doi.org/10.1109/TVCG.2013.267

[8] Schuller, C.; Panozzo, D.; Sorkine-Hornung, O.: Appearance-mimicking surfaces, ACM Transactions on Graphics, 33(6), 2014, 216:1–216:10.

[9] Zhang, Y.W.; Zhang, C.; Wang, W.; Chen, Y.: Adaptive bas-relief generation from 3d object under illumination, Computer Graphics Forum, 35(7), 2016, 311–321. https://onlinelibrary.wiley.com/doi/full/10.1111/cgf.13028

[10] Zhang, Y.W.; Qin, Bb.; Chen, Y.; Ji, Z.; Zhang, C.: Portrait relief generation from 3d object, Graphical Models, 102, 2019, 10–18. https://doi.org/10.1016/j.gmod.2019.01.002

[11] Wei, M.; Tian, Y.; Pang, W.M.; Wang, C.C.L.; Pang, M.Y.; Wang, J.; et al.: Bas-relief modeling from normal layers, IEEE Transactions on Visualization and Computer Graphics, 102(4), 2019, 1651–1665. https://doi.org/10.1109/TVCG.2018.2818146

[12] Ji, Z.; Feng, W.; Sun, X.; Qin, F.; Ma, W.: Reliefnet: Fast bas-relief generation from 3d scenes, Computer-Aided Design, 130, 2020, 102928. https://doi.org/10.1016/j.cad.2020.102928

[13] Arpa, S.; Susstrunk, S.; Hersch, R.D.: Sculpture paintings, In: Proceedings of the Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering, ACM, 2018.

[14] Botsch, M.; Sorkine, O.: On linear variational surface deformation methods, IEEE Transactions on Visualization and Computer Graphics, 14(1), 2008, 213–230. https://doi.org/10.1109/TVCG.2007.1054

[15] Sorkine, O, Botsch, M. Interactive Shape Modeling and Deformation. In: Museth, K, Weiskopf, D.; editors. Eurographics 2009 - Tutorials. The Eurographics Association; 2009.

[16] Sorkine-Hornung, O.; Alexa, M.: As-rigid-as possible surface modeling, In: Eurographics Symposium on Geometry Processing, 2007, 109–116.

[17] Ju, T.; Schaefer, S.; Warren, J.D.: Mean value coordinates for closed triangular meshes, ACM Transactions on Graphics, 2005, 24(3), 561–566. http://dl.acm.org/doi/10.1145/1186822.1073229

[18] Lipman, Y.; Levin, D.; Cohen-Or, D.: Green coordinates, ACM Transactions on Graphics, 2008, 27(3), https://doi.org/10.1145/1360612.1360677

[19] Yifan, W.; Aigerman, N.; Kim, V.G.; Chaudhuri, S.; Sorkine-Hornung, O.: Neural cages for detail-preserving 3d deformations. In: CVPR. 2019, 72–80. https://arxiv.org/abs/1912.06395

[20] Zhang, Y.W.; Zhang, C.; Wang, W.; Chen, Y.; Ji, Z.; Liu, H.: Portrait Relief Modeling from a Single Image, IEEE Transactions on Visualization and Computer Graphics, 26(8), 2020, 2659-2670. https://doi.org/10.1109/TVCG.2019.2892439

[21] Leifman, G.: Shtrom, E.; Tal, A.: Surface regions of interest for viewpoint selection, In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, 414–421. https://doi.org/10.1109/TPAMI.2016.2522437