www.cadanda.com
Computer-Aided Design™
AND APPLICATIONS

Modeling Personalized Lampshade with Lithophane

Jinlei Wang, Wenfei Long, Yanzhao Chen and Yu-Wei Zhang

School of Mechanical and Automotive Engineering, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China

Corresponding author: Yu-Wei Zhang, zhangyuwei_scott@126.com

Abstract. Lithophane lampshade is an artwork that has been commonly used in home decorations. Traditional lithophane lampshade creation requires a lot of user interactions, which is time-consuming. In this paper, we present an automatic method for personalized lithophane lampshade modeling. Given a 2D target image and a personalized 3D lampshade surface, our method first applies Laplacian-based mesh deformation to flatten the surface, and establishes mapping relation with the target image. Then, Poisson-based surface reconstruction is used to generate the lithophane. In the end, an optimization step is applied to adjust the coordinates of the lithophane. In our modeling framework, users are allowed to freely adjust the input source and the thickness of the lampshade. Then, our method automatically output the modeling result in a short period of time. Experiments with a range of models show that our method is effective in modeling lithophane lampshade with target image shown under lighting.

Keywords: lithophane, lampshade modeling, mesh deformation, surface flattening.
DOI: https://doi.org/10.14733/cadaps.2020.850-860

1 INTRODUCTION

A lithophane lampshade is an etched or molded artwork with lithophane textures attached on the outer surface, presenting interesting 2D image with varying shades of grey when back lit by a light source, as shown in Figure 1. Traditional lithophane lampshades were made by specialized craftspeople beginning as an image carved in translucent materials, relying much on experienced skills. Currently, computer-assisted modeling facilitates the design of lithophane lampshade, but a large number of user interactions are still required. Moreover, the input lampshades in most software platforms are only limited to simple shapes such as planes, cylindrical surface and spherical surface et al.

This paper focuses on the automatic creation of lithophane lampshade with personalized curved shapes. Given a target 2D image and 3D curved surface as input, our target aims to convert the image into the form of 2.5D lithophane, which can then present image of the target under lighting. To this end, we first use Laplacian-based mesh deformation to flatten the lampshade surface. Through image mapping, we then convert the pixel grey-values into the offset values for lithophane creation. Taking the offset values as constraints, we construct the lithophane by Poisson-based surface reconstruction, followed by an optimization step to adjust the lithophane coordinates. Finally, the full 3D model of the lampshade is generated and output for fabrication. In our modeling framework, users are required to specify the minimum and maximum thickness of the lampshade according to the translucency of the materials. Once the thickness parameters are given, our method is able to automatically generate a desired lampshade model in a short period of time.

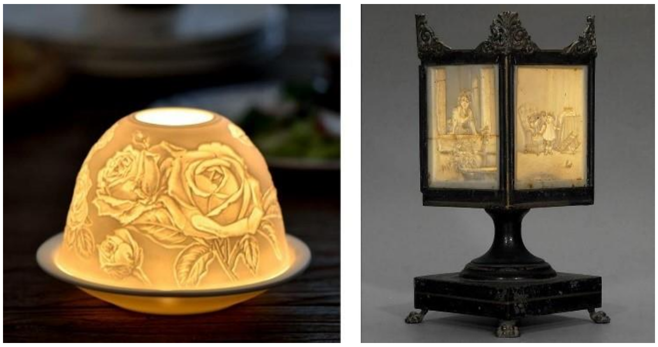

Figure 1: Personalized lithophane lampshade.

We validate our method by a number of experiments. The results show that our method is capable of generating lampshade models with desired lithophane. Furthermore, we test the physical models fabricated by 3D printing under lighting. The results indicate the effectiveness of our proposed method. The work of this paper is organized as follows. Related works are described in Section 2. Modeling pipeline is introduced in section 3. In section 4, we show experimental results and discussions. Section 5 gives conclusions and potential future works.

2 RELATED WORKS

Lithophane can be considered as a special type of bas-relief attached on the outer surface of the lampshade. In the following, we briefly introduce state-of-the-art techniques on computer-assisted bas-relief modeling.

Cignoni et al. [5] pioneered the research of relief modeling from a 3D object. They used perspective transformation to directly compress the model, which failed to preserve the geometric details of the model. Weyrich et al. [16] nonlinearly compressed the gradient magnitude of the input height field and then used Poisson equation to reconstruct a new height field, improving the quality of generation. Bian et al. [2] also compressed the height filed in gradient domain, but recovered geometrical details in object domain. They added details back to the relief surface when the height field was reconstructed. Zhang et al. [26] formulated bas-relief modeling as a problem of gradient-based mesh deformation, which directly performs height compression and details preservation. Ji et al. [10] used a normal image as input to generate continuous relief surface with similar appearance to the input. Inspired by this work, Wei et al. [20] and Ji et al. [11] respectively introduced decomposition-and-composition frameworks to handle the input normal image, further enhancing the relief details. Similarly, Schüller et al. [14] presented a mathematical framework to construct relief surface, approximating the normals of a given 3D shape, while strictly obeying given depth constraints. Through normal manipulation, Zhang et al. [27] also proposed an adaptive framework to produce bas-reliefs with respect to illumination conditions.

Compared with 3D objects, images are much easier to capture, indicating wider applications. Alexa et al. [1] applied Shape from Shading (SFS) to create bas-reliefs from single images. Wu et al. [17] proposed to generate bas-reliefs from frontal photographs of human faces. Given a portrait image, their method first generated a new image corresponding to the relief appearance. Then the relief surface was constructed by SFS. In their subsequent work [18], they combined non-photorealistic rendering (NPR) technology into previous framework, improving the relief visual effect. Zeng et al. [24] proposed a region-based approach to generate relief. They first generated a base surface from region layers, and then used the gray and gradient information from the image to optimize the relief surface. Li et al. [12] presented a two-level framework to restore Brick-and-Stone Relief (BSR) from their rubbing image. Zhang et al. [25] constructed stylized Chinese calligraphy relief by combining a homogeneous and an inhomogeneous height field via nonlinear compression.

There are also some literatures on lithophane-like products creation. Brooks et al. [4] took a 2D image and a 3D surface as input to create personalized lampshades. Their method can be applied to 3D surface with general shapes such as planes and cylindrical surface, but not applicable to undevelopable surfaces. Song et al. [15] introduced an image enhancement algorithm to improve lithophane details. Weiler et al. [19] presented a Lithobox system that use ceramic and lighting technique to create illuminated 3D models. Mitra et al. [13] constructed art sculptures with 3D shadow. Under illumination, the sculpture model can project the contour of target images to achieve imaging effects. Zhao et al. [23] constructed 3D printed lampshades by a novel halftone technology. Yang et al. [21] discussed the problem of engraving binary images on the target surface, and then proposed a binary image optimization framework to make the engraving model retain more details. Zhou et al. [22] synthesized decorative patterns on curved shapes, and users can specify the patterns thickness to create 3D printed models. Similarity, Chen et al. [9] constructed attractive 3D models with simple graphics attached on the three-dimensional base surface.

In contrast to the above approaches that create bas-reliefs or lithophane-like surfaces from 3D objects or 2D images, we aim to automatically generate lithophane on personalized curved lampshades in this paper. Different from most previous approaches, our method takes 3D mesh deformation as modeling basis, which proves to be an effective way in modeling lithophane lampshades with target images shown under lighting.

3 METHOD

3.1 Overview of the Pipeline

Our method takes a target image and a personalized lampshade surface as input, and output a full lampshade model with lithophane on the outer surface. The whole modeling pipeline consists of four main steps, shown in Figure 2.

- **Preprocessing.** A triangular lampshade surface and a target image are prepared as the input.
- **Surface flattening.** To facilitate image mapping, the lampshade surface is flattened into a rectangle plane using Laplacian-based mesh deformation.
- **Image mapping.** The target image is mapped on the flattened plane, and a linear function is formulated to convert the grey values on the target image into the offset values for lithophane creation.
- **Lampshade modeling.** Lithophane is generated using Poisson-based reconstruction, and triangles at the top and bottom of the lampshade are filled to form a full 3D lampshade model.

3.2 Preprocessing

Before lithophane creation, users are required to provide a 2D target image and a 3D lampshade surface, which might be modeled in any kind of 3D modeling software such as Unigraphics NX [6] and Creo Parametric [7]. After that, HyperMesh [8] is used to convert the input3D surface into the form of triangular mesh. To ensure the mapping accuracy afterwards, we restrict the number of mesh vertices to be higher or equal to the number of image pixels.

3.3 Surface Flattening

To facilitate the mapping of target image onto the 3D lampshade surface, we flatten the lampshade surface into the form of 2D rectangle plane. For the undevelopable freeform surface, distortions are unavoidable after flattening operations. To reduce the distortions and improve the visual effects of modeling, we propose the following flattening algorithm.

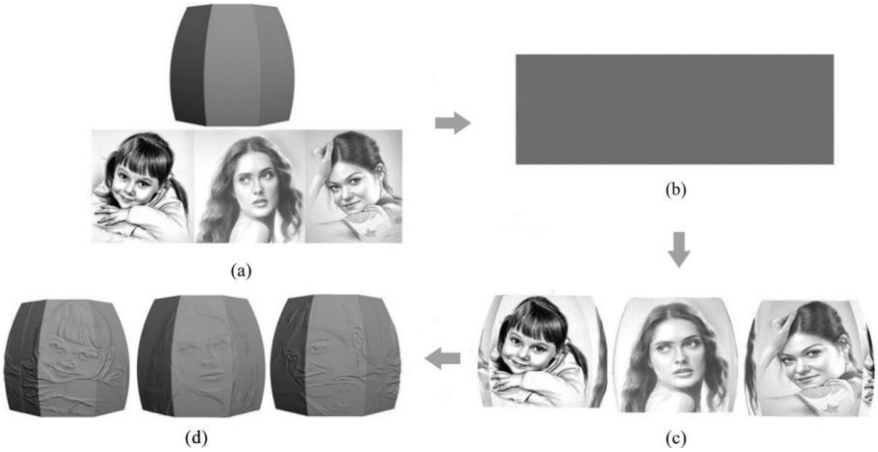

Figure 2: Pipeline of the proposed method (a) input lampshade surface (b) flattened lampshade plane (c) texture mapping (d) full 3D lampshade model with lithophane generated on the outer surface.

3.3.1 *Cutting line definition*

We assume the lampshade surfaces designed by users are open in y-direction, e.g. the input surface has two border loops at the top and bottom. As the lampshade is still closed in the x- direction, we now define a boundary line to open the surface. In doing so, we automatically detect the vertical edge on back side of the lampshade (negative direction of z- axis) and use it as the cutting line. The one-ring neighbor triangles around the cutting line will then be removed. As a result, the input surface becomes open both in x- direction and y- direction, as shown in Figure 2a.

3.3.2 *x- direction flattening*

Excessive distortions will be produced inside the surface if only vertices on the left and right edges of the cutting line are used as constraints to flatten the surface in x- direction. To solve this problem, we further select six vertices (p_5, p_6, p_7, p_8, p_9 and p_{10}) on the top and bottom edges as auxiliary constraints. These constraint vertices are automatically detected on leftmost, rightmost and front of upper and lower boundaries, as shown in Figure 3.

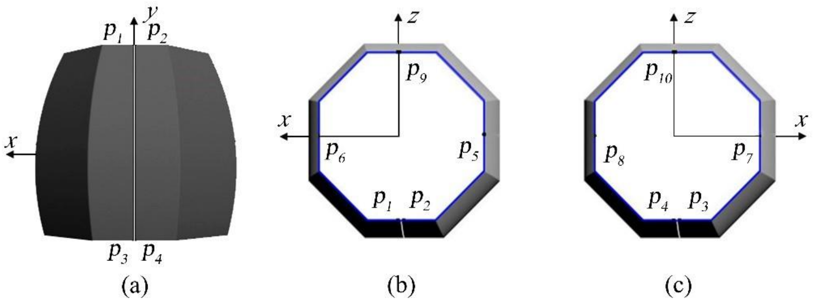

Figure 3: Defining a cutting line on the input surface (a) back view of the surface, in which a cutting line has been defined on the center (b) top view of the surface (c) bottom view of the surface.

We flatten the input surface in x- direction by solving the following Laplacian-based linear system:

$$L\cdot X=b \tag{3.1}$$

where L is the Laplacian matrix, b is the divergence vector, and X is the x- coordinates need to be computed. When defining the divergence vector, it is assumed that each triangle rotates around its center until its normal to be aligned to z axis. Then, each vertex divergence can be calculated by gradients of the neighboring triangles. For more details about the definition of divergence, please refer to [3].

Before solving Eq.1, we need to compute the lengths of upper and lower boundaries by chord length accumulation:

$$S_a=\sum_{i=0}^{m}\sqrt{(x_{i+1}\text{-}x_i)^2+(z_{i+1}\text{-}z_i)^2}\ ,\ S_b=\sum_{i=0}^{n}\sqrt{(x_{i+1}\text{-}x_i)^2+(z_{i+1}\text{-}z_i)^2} \tag{3.2}$$

where m and n are the number of vertices on the boundaries from the start points p_1 and p_3 to the end points p_2 and p_4. When solving Eq.1, the x- coordinates of the constraints should be given in advance. In doing so, we fix the x- coordinates of the left boundary vertices to be $-(S_a+S_b)/4.0$, and the right boundary vertices to be$(S_a+S_b)/4.0$.The auxiliary constraint vertices $p9$ and $p10$ are on the center of the rectangle plane, with their x- coordinates to be zero. The x- coordinates of p_5 and p_8 are set to be identical, whose values equal to $-(S_a+S_b)/8.0$. Similarly, the x- coordinates of p_6 and p_7 equal to $(S_a+S_b)/8.0$.

3.3.3 y- direction flattening

We flatten lampshade surface in y- direction by solving the following Laplacian-based linear system:

$$L\cdot\Delta y=0 \tag{3.3}$$

where L is the Laplacian matrix, Δy is the incremental coordinates need to be solved. Similar to Eq.1, we compute the lengths of left and right boundaries by chord length accumulation:

$$S_l=\sum_{i=1}^{s}\sqrt{(y_{i+1}\text{-}y_i)^2+(z_{i+1}\text{-}z_i)^2},\quad S_r=\sum_{i=1}^{k}\sqrt{(y_{i+1}\text{-}y_i)^2+(z_{i+1}\text{-}z_i)^2} \tag{3.4}$$

where s and k are vertex numbers on the boundaries. The y- coordinate increments of upper and lower boundary vertices are used as the boundary conditions when solve Eq.3. Their values are set to be$(S_l+S_r)/4.0\text{-}y_i$, where y_i is y- coordinates of the boundary vertices.

After solving Eq.3, the final y- coordinate for each vertex on the lampshade surface are updated by:

$$y_i^*=\Delta y_i+y_i \tag{3.5}$$

Figure 4 shows variations of the triangle areas after surface flattening, where red colors indicate maximum values, and blue colors indicate minimum values. It can be seen that flattening with auxiliary constraints can significantly reduce the deformation distortions, which is important to ensure the visual quality of modeling.

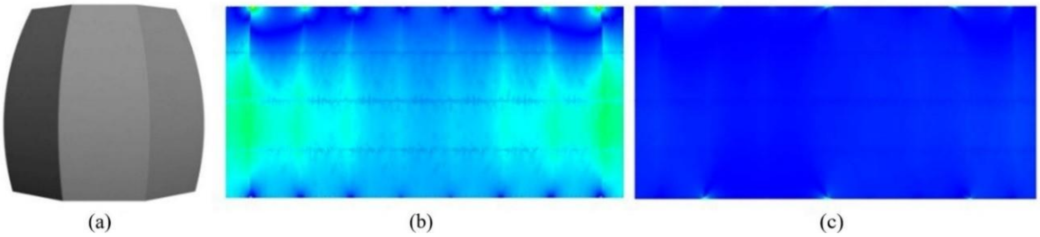

Figure 4: Surface flattening (a) input lampshade surface (b) variations of the triangle areas without auxiliary constraints (c) variations of the triangle areas with auxiliary constraints.

3.4 Image Mapping

As the lampshade can only present grayscale images when lit by a lighting source, according to the convention, we convert the input RGB image into a 8-bit grayscale image by: $h'=(0.2989\times R+0.5870\times G+0.1140\times B)$, and normalize the gray value by: $h=h'/255$. Then, the scaling ratio between the input image and the flattened surface are calculated by:

$$\alpha=min(\frac{W}{L},\ \frac{H}{D}) \tag{3.6}$$

where W and H are the number of pixels in x- and y- directions respectively, L and D are width and height of the rectangle plane. By aligning the center of the grayscale image with the center of the flattened plane, we can get the pixel coordinates for each vertex:

$$x_{dot}=R(x*\alpha+\frac{W}{2}),\quad y_{dot}=R(\frac{H}{2}-y*\alpha) \tag{3.7}$$

Where x and y are vertex coordinates on the flattened plane, x_{dot} is column pixels number and y_{dot} is row pixels number, α is the scaling ratio and R represents the round operator. Figure 5 shows an example of the mapping result, with 2D textures mapped on the 3D lampshade surface.

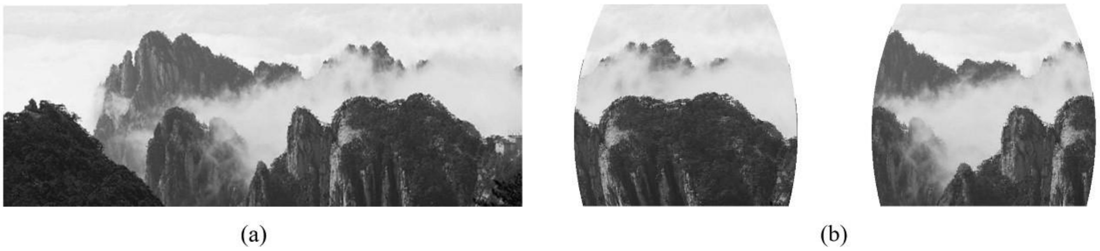

Figure 5: Image mapping (a) input target image (b) surface with mapped image.

After image mapping, we compute the offset for each vertex on the lampshade surface by:

$$d_i=T_{min}+(T_{max}-T_{min})\frac{G_{max}-G_i}{G_{max}-G_{min}} \tag{3.8}$$

where d_i is the offset value, T_{max} and T_{min} are maximum and minimum thickness of the lampshade in terms of the translucency of the material, G_i is the corresponding gray value, G_{max} and G_{min} are maximum and minimum gray values in the target image.

3.5 Lampshade Modeling

In this section, we generate a full lampshade model with lithophane attached on the outer surface. If the input surface is directly offset outward along the normal directions, self-intersections might be produced on local points that have large curvatures, resulting artifacts on the lithophane surface, as shown in Figure 6. To avoid the artifact, we first offset each triangle on the lampshade surface along its normal to a distance estimated by the average offset value of the vertices. Then, we construct the lithophane by solving the following Poisson-based equations:

$$L*x=b_x,\quad L*y=b_y,\quad L*z=b_z \tag{3.9}$$

where L is the Laplacian operator, x, y and z are vertex coordinates of the lithophane need to be solved, b_x, b_y and b_z are divergence vectors in x-, y- and z- directions.

To solve Eq.9, we take odd vertices on the upper and lower boundaries as constraints, whose coordinates are predefined as:

$$x_i'=x_0+n_x*d_i,\quad y_i'=y_0+n_y*d_i,\quad z_i'=z_0+n_z*d_i \tag{3.10}$$

where x_0, y_0 and z_0 are coordinates of the vertices on the input surface, n_x, n_y and n_z are components of vertex normal, and d_i is the offset value of the constraint vertex.

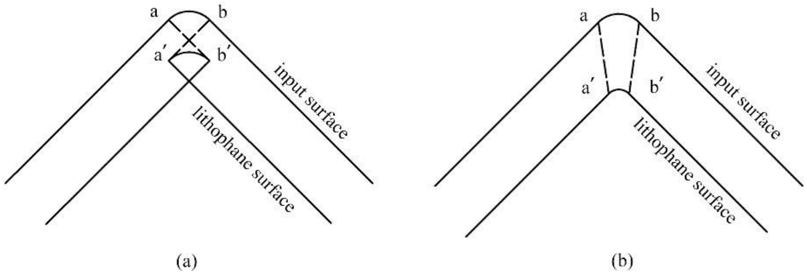

Figure 6: Self-intersections might be produced on large-curvature points (a) straightforward offsetting (b) Poisson-based surface reconstruction.

As the outer lithophane surface has the same topology with the input surface, we are able to adjust the offset directions by:

$$e_x=x\text{-}x_0,\ e_y=y\text{-}y_0,\ \ e_z=z\text{-}z_0 \tag{3.11}$$

where x, y and z are vertex coordinates on the resulting lithophane solved by Eq.9, x_0, y_0 and z_0 are coordinates of the corresponding vertex on the input surface, e_x, e_y and e_z are components of the new offset directions. Since Poisson-reconstructed lithophane surface does not strictly guarantee the offset values deduced by Eq.8, we optimize the vertex coordinates by:

$$x_0^*=x_0+e_x*d_i,\ \ y_0^*=y_0+e_y*d_i,\quad z_0^*=z_0+e_z*d_i \tag{3.12}$$

where d_i is the offset value computed by Eq.8.

Once the lithophane coordinates have been optimized, we finally fill triangles between the input surface and the lithophane surface at the top and bottom, generating a closed 3D lampshade model, as shown in Figure 7.

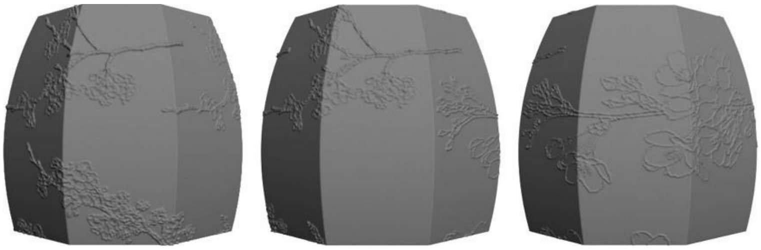

Figure 7: Full 3D lampshade model from different viewing directions.

4 EXPERIMENTAL RESULTS

4.1 Performance

We implemented the proposed method using C++ and tested it on a PC with an Intel(R) Core (TM) i5-3230M CPU @ 2.6 GHz and 2.88 GB RAM. In our experiments, the resolution of the target image is set to 500×800 and the triangular input surface has about 400k vertices.

In the preprocessing stage, it takes about 5~10 minutes for a user to design a personalized lampshade surface in 3D modeling software, and 30 seconds to remesh the input surface in HyperMesh. Taking the mesh surface and the target image as input, our proposed method takes about 10 seconds to flatten the lampshade surface, and another 20 seconds to model the lampshade model.

4.2 Modeling Results

Models with varying thickness. In our modeling framework, users are allowed to adjust the lampshade thickness by altering parameters T_{max} and T_{min} in Eq.8 according to the translucency of the fabricating material. We use photosensitive resins as the fabrication material in our experiments. According to the lighting attribute of photosensitive resins, the lampshade thickness should be linear with the shadings produced by lights passing through the material between 0.7 and 2.4 millimeters. Therefore, the thickness between the inner input surface and the outer lithophane surface should be strictly limited in this range. Considering the minimum thickness of 3D printing, we set T_{min} to be 1.0 millimeter by default. Figure 8 shows two examples of the resulting models with different thickness. It can be seen that the details of the lithophane on both models have been well constructed.

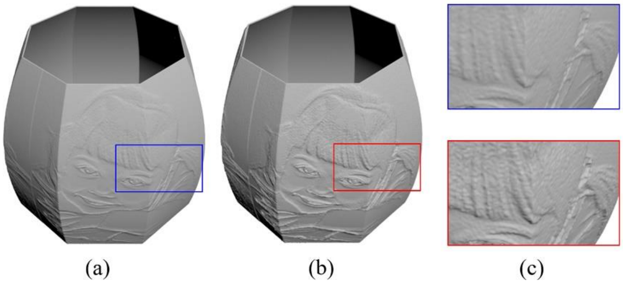

Figure 8: Lampshades with varying thickness (a) T_{min}=2.0 mm (b) T_{max}=3.0 mm (c) enlarged views of the lithophane.

Models with different input target images. In our modeling framework, users can freely select their own target images and use them as inputs. Then, our method automatically generates lithophane on the lampshade. Figure 9 shows several resulting models with different input images mapped onto the same lampshade model.

Models with varying input surfaces. Theoretically, any freeform lampshade surface can be flattened and aligned with the input image. In this paper, we limit the input lampshade surface to the rotational type that has upper and lower boundaries because they are the most popular types in real applications. Figure 10 shows the modeling results with different types of shapes. It is noted that our method can not only model lithophane lampshades with complex shapes, but also can be used to generate lampshades with general shapes, as shown in Figure 11.

Fabrication. To verify the lighting effects, we fabricated some lampshade models by 3D printing using photosensitive resins. In our experiments, the height of the lampshade model is set to be 200 mm, the maximum and minimum thicknesses are 1.0 millimeters and 2.4 millimeters respectively, and we used LEDs as the lighting source. As shown in Figure 12, lampshade models presented target images with fine shading details under lighting.

5 CONCLUSIONS AND FUTURE WORKS

This paper presents an automatic method for personalized lithophane lampshade creation. It takes a 2D image and a 3D curved surface as input and outputs a full 3D lampshade model with lithophane, in which only a few user interactions are required. The fabricated models presented desired images under lighting. Our method can not only be applicable to lampshade creation with general surface such as cylindrical, conical and spherical surface, but also applicable to the models with complex shapes. Once the target image and the personalized surface are given, our method automatically creates lithophane in a short period of time. The experimental results verify the effectiveness of our method.

In our implementation, we convert the input color image into a grayscale image using a straightforward interpolation method, which might not adequate to preserve fine details in the lithophane. In the future, more feature-preserving method should be explored, so that the lampshade can better present image details under the lighting.

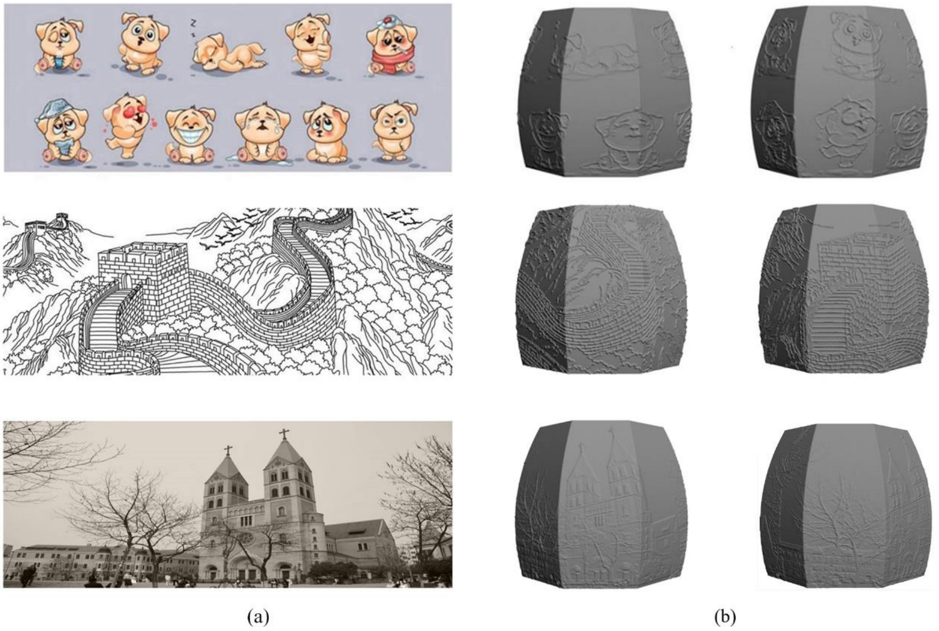

Figure 9: Lampshades with different input images (a) input images (b) models with different target images mapped on the outer surface (thickness range: 1.0~2.4 mm).

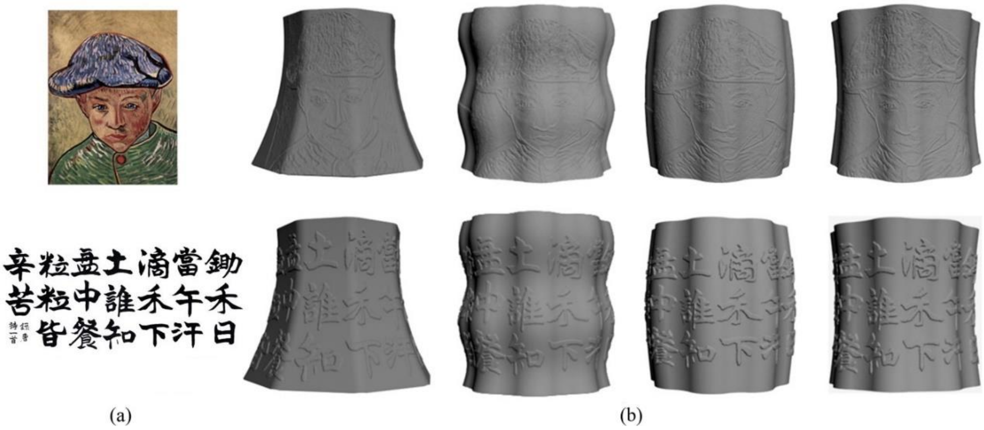

Figure 10: Models with varying shapes (a) target images (b) results with varying shapes (thickness range: 1.0~2.4 mm).

ACKNOWLEDGEMENTS

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

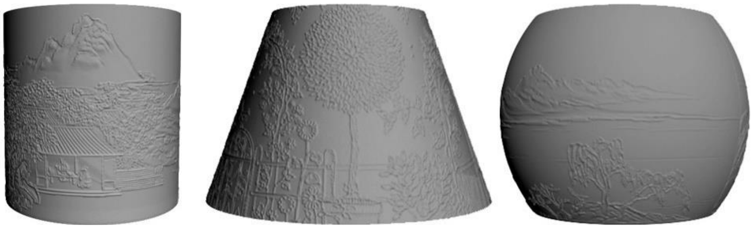

Figure 11: Models with cylindrical, conical and spherical shapes (thickness range: 1.0~2.4 mm).

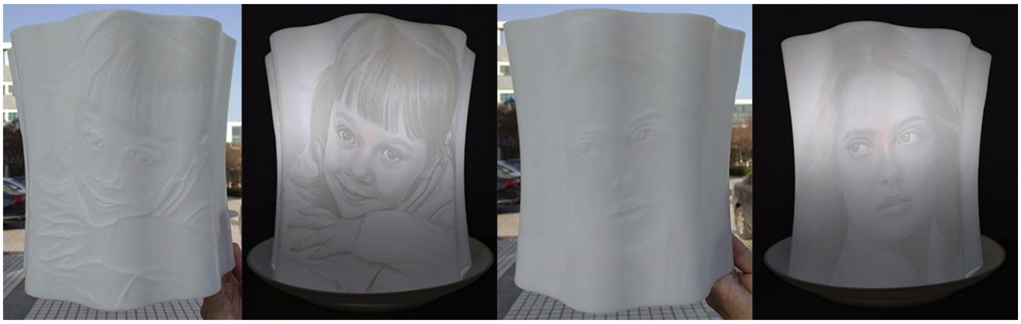

Figure 12: Physical models fabricated by 3D printing and lit by LEDs source.

Yanzhao Chen, https://orcid.org/0000-0002-6517-527X
Yu-Wei Zhang, https://orcid.org/0000-0001-6566-5714

REFERENCES

[1] Alexa, M.; Matusik, W.: Reliefs as images, ACM Transactions on Graphics, 29(4), 2010, No.60. http://doi.org/10.1145/1778765.1778797

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

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

[4] Brooks, H.; Ulmeanu, M.; Abram, T.: Production of personalized lithophane lighting products using additive manufacturing, 4th International Conference on Additive Technologies,2012, 1-10. doi: 10.2507/daaam.scibook.2012.xx

[5] Cignoni, P.; Montani, C.; Scopigno, R.: Computer-assisted generation of bas- and high-reliefs, Journal of Graphics Tools, 2(3), 1997, 15-28. http://doi.org/10.1080/10867651.1997.10487476

[6] Unigraphics NX: https://www.siemens.com/nx

[7] Creo Parametric: https://www.ptc.com

[8] HyperMesh: https://altairhyperworks.com.cn/

[9] Chen, W.; Zhang, X.; Xin, S.; Xia, Y.; Lefebver, S.; Wang, W.: Synthesis of filigrees for digital fabrication, ACM Transactions on Graphics, 35(4), 2016, No.98. http://doi.org/10.1145/2897824.2925911

[10] Ji, Z.; Ma, W.; Sun, X.: Bas-relief modeling from normal images with intuitive styles, IEEE Transactions on Visualization and Computer Graphics, 20(5), 2014, 675–685. http://doi.org/10.1109/TVCG.2013.267

[11] Ji, Z.; Sun, X.; Ma, W: Normal image manipulation for bas-relief generation with hybrid styles, 2018. arXiv:1804.06092

[12] Li, Z.; Wang, S.; Yu, J.; Ma, K. L.: Restoration of brick and stone relief from single rubbing images, IEEE Transactions on Visualization and Computer Graphics, 18(2), 2012, 177–187. http://doi.org/10.1109/TVCG.2011.26

[13] Mitra, N. J.; Pauly, M.: Shadow art, ACM Transactions on Graphics, 28(5), 2009, No.156. http://doi.org/10.1145/1661412.1618502

[14] Schüller, C.; Panozzo, O.: Appearance-mimicking surfaces, ACM Transactions on Graphics, 33(6), 2014, No. 216. http://doi.org/10.1145/2661229.2661267

[15] Song, T.; Xing, Z.: A lithophane model generation method based on gray-scale compression, Computer Engineering, 43(10), 2017, 253-258. http://doi.org/10.3969/j.issn.1000-3428.2017.10.042

[16] Weyrich, T.; Deng, J.; Barnes, C.; Rusinkiewic, S.; Finkelstein, A.: Digital bas-relief from 3D scenes, ACM Transactions on Graphics, 26(3), 2007, 32-38. http://doi.org/10.1145/1276377.1276417

[17] Wu, J.; Martin, R. R., Rosin, P. L.; Sun, X.-F.; Langbein, F.-C.; Lai, Y.-K.; Marshall, A. D.; Liu, Y.-H.: Making bas-reliefs from photographs of human faces, Computer-Aided Design, 45(3), 2013, 671–682. http://doi.org/10.1016/j.cad.2012.11.002

[18] Wu, J.; Martin, R. R.; Rosin, P. L.; Sun, X.-F.; Lai, Y.-K.; Liu, Y.-H.; Wallraven, C.: Use of non-photorealistic rendering and photometric stereo in making bas-reliefs from photographs, Graphical Models, 76(4), 2014, 202–213. http://doi.org/10.1016/j.gmod.2014.02.002

[19] Weiler, J.; Piyum, F.; Todd, I.; Stacey, K.: Lithobox: Creative practice at the intersection of craft and technology, ACM TEI '19 Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction, 2019, 471-477. http://doi.org/10.1145/3294109.3301258

[20] Wei, M.; Tian, Y.; Pang, W.; Wang, C.; Pang, M.; Wang, J.; Qin, J.; Heng, P.: Bas-relief modeling from normal layers, IEEE Transactions on Visualization and Computer Graphics, 25(4), 2018, 1651-1665. http://doi.org/10.1109/TVCG.2018.2818146

[21] Yang, J.; He, S.; Lu, L.: Binary image carving for 3D printing, Computer-Aided Design, 114, 2019, 191-201. http://doi.org/10.1016/j.cad.2019.05.028

[22] Zhou, S.; Jiang, C.; Lefebver, S.: Topology-constrained synthesis of vector patterns, ACM Transactions on Graphics, 33(6), 2014, No.215. http://doi.org/10.1145/2661229.26 61238

[23] Zhao, H.; Lu, L.; Wei, Y.; Lischinski, D.; Sharf, A.; Cohen-or, D.; Chen, B.: Printed perforated lampshades for continuous projective images, ACM Transactions on Graphics, 35(5), 2016, No.154. http://doi.org/10.1145/2907049

[24] Zeng, Q.; Martin, R. R.; Wang, L.; Quinn, J. A.; Sun, Y.; Tu, C.: Region-based bas-relief generation from a single image, Graphical Models, 76(3), 2013, 140–151. http://doi.org/10.1016/j.gmod.2013.10.001

[25] Zhang, Y.-W.; Chen, Y.; Liu, H.; Ji, Z.; Zhang, C.: Modeling Chinese calligraphy reliefs from one image, Computers and Graphics, 70, 2018, 300–306. http://doi.org/10.1016/j.cag.2017.07.022

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

[27] 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. http://doi.org/10.1111/cgf.13028