**Bas-Relief Generating from Given 3D Model**

CS:995 – Masters Capstone project

Spring 2017

by

Lei Yao

under the supervision of

Professor Dr. Zeyun Yu

**Abstract**

Traditional 3D model-based bas-relief modeling methods are often limited to model-dependent and monotonic relief styles. Our project uses a novel method for digital bas-relief modeling with intuitive style control. Given a 3D model, the problem we solve involves generating a discontinuity-free depth field with high compression of depth data while preserving or even enhancing fine details.. The bas-relief style is controlled by choosing a parameter and setting a targeted height for them. Bas-relief modeling and stylization are achieved simultaneously by solving a sparse linear system. Experiments with a wide range of 3D models and scenes show that our method can effectively generate digital bas-reliefs.

**Introduction**

We present a novel method based on given 3D-model files. First step of our project is generate corresponding and high quality normal maps from 3D-model. Second step is to develop a flexible method for bas-relief generation which clearly preserves or even enhances visible shape details.

**Background**

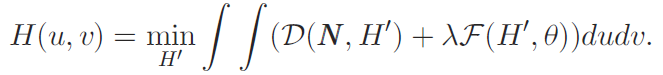
Digital bas-relief generation is a young research topic in the computer graphics field. It is well known that normal play an important and essential role in real-time rendering. Normal is also essential in the inverse problems of rendering, namely photometric stereo and shape from shading.

Normals are also important in 3D modeling. This time our goal is to create bas-reliefs via solving the fundamental problems which include preserving the appearance for orthogonal views and squashing the discontinuity gaps. Mainly procedure is taking 3D models as input, then acquiring the normal map from a perspective. Next, simply treat the normal map and go further generate bas-relief from that normal map.

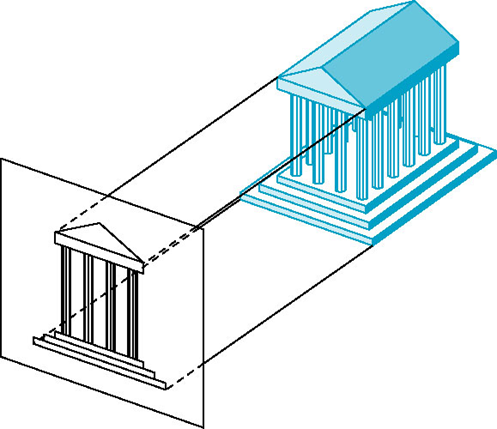
**Bas-relief Modeling and Stylization**

**Algorithm Overview**

Bas-relief presents the unique challenge of squeezing shapes into a nearly-flat surface while maintaining the fine details of the 3D scene as much as possible.[2] So, the important thing is not only construct a relatively flatten surface, but also preserve the whole detail from the object.

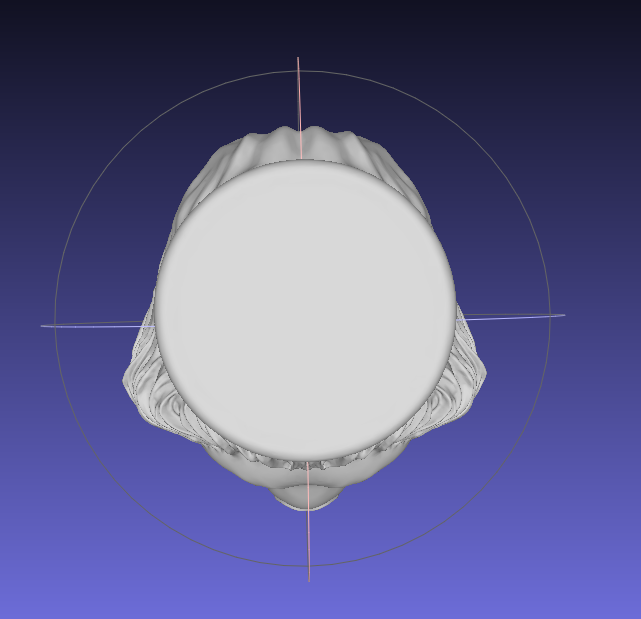
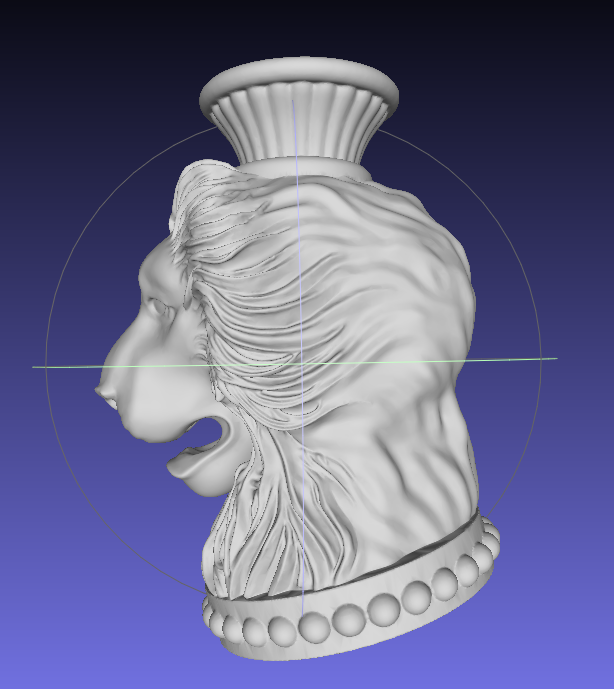
This time, by reading the paper, our idea is generate height of each pixel at the surface with the help of normal vectors. Here is the formula:

This energy function is a sum of two terms: one that measures the detail similarity of N and H0 (D means ‘Detail’), and the second one tries to fix a height constraint (F means ‘Flatten’). Theta is a threshold used to control the height of the resulting bas-relief. The relative contributions of two items are controlled by the parameter phi.[1]

**Normal Image Generation**

**Figure 1 Choose a perspective and generate corresponding picture**

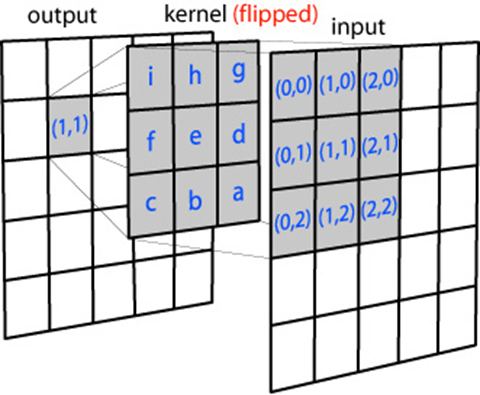
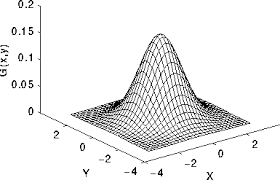
By reading a 3D Model (OFF files), we can generate a normal map from the perspective user select. Because actually, a 3D model is consist of a certain number of triangles. By calculating the cross product of each triangle, their normal vectors can be attained. Then using ray-trace algorithm to shot out a ray from each pixel, if the ray touch with input 3D model, set that pixel’s RGB value respectively each to the normal vector’s x, y, z values. If it doesn’t touch anything, just set its color be blue (0,0,1).



1. **(b) (c)**

**Figure 2 Three views of lion head 3D model, from left to right followed by the main view, side view, plan view.**

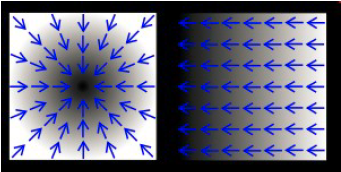
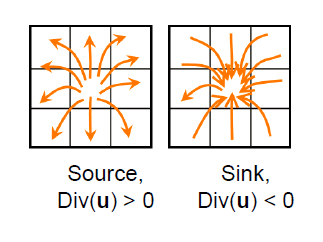
3D model is three-dimensional, so if the user want to generate a bas-relief from a rotated perspective, we can rotate it first then calculate and obtain normal values.

****

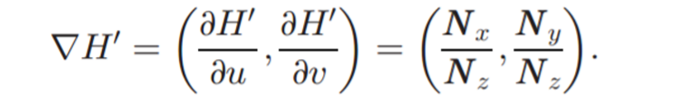
**Figure 3 Gaussian Filter, construct a template to calculate one point and its neighbors in order to smooth the value and eliminate noise. (\*picture from: http://homepages.inf.ed.ac.uk/rbf/HIPR2/figs/gauss2.gif)**

In case of there exist some noise due to the some mistakes when gathering normal vectors. It’s a fair step to treat these data by using Gaussian filer, after that, high quality data can be guaranteed for next step.

**Laplacians from Normals**

 ****

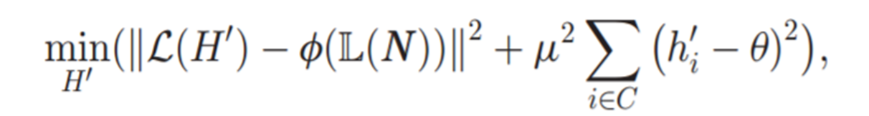
**Figure 4 Describe the divergence, if one point’s divergence is a positive number, it means it is a source point, otherwise, it is a sink point.**

Laplace equation play an important role in height calculation part. After obtaining normal vectors, calculating the second derivative value based normal values for each pixel. Here, the second derivative namely is divergence’s Laplacian equation of each pixels. In figure 5, left picture is the normal map and right hand side is the divergence map.

****

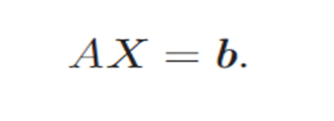
**Figure 5 Normal map of lion head and LN image of lion head.**

From Laplacian of divergence image, it preserve the whole feature of lion head picture. Then, next is the critical part, taking advantage of the Laplacian equation, let the height of each pixel has the same feature as the divergence. So, original formula should be transferred as:



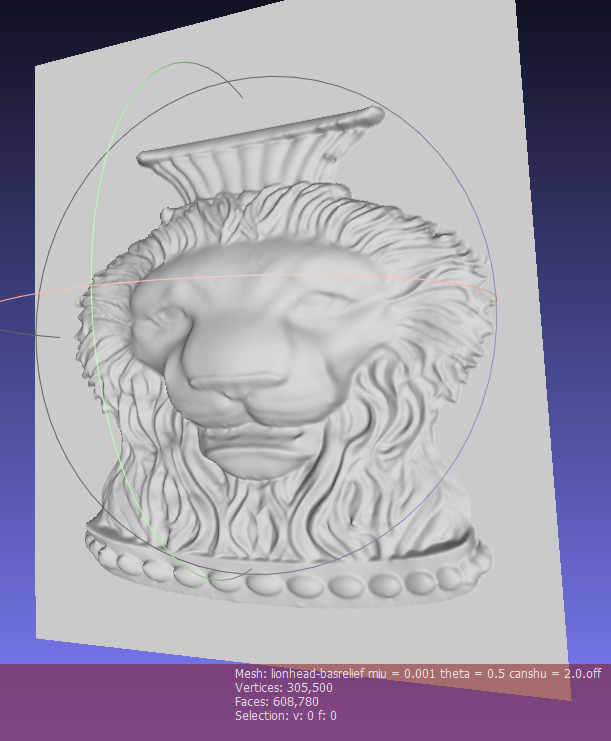
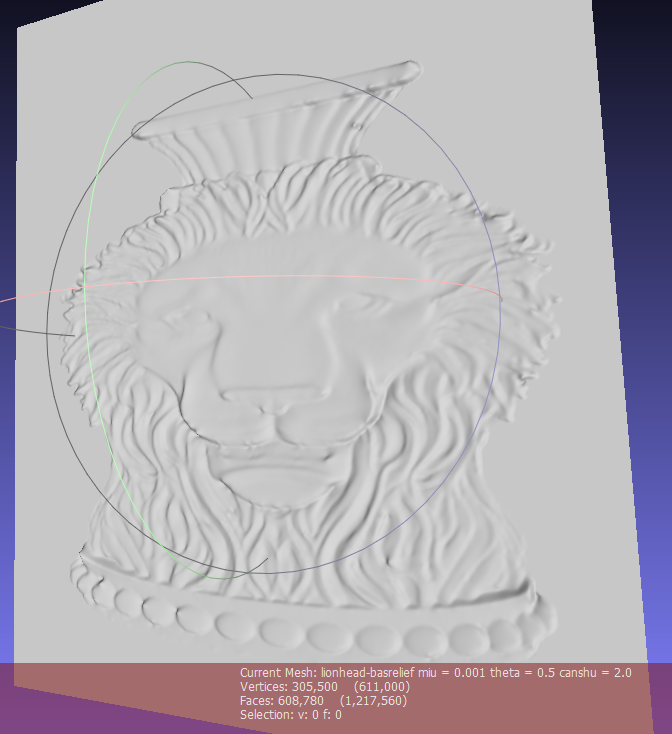
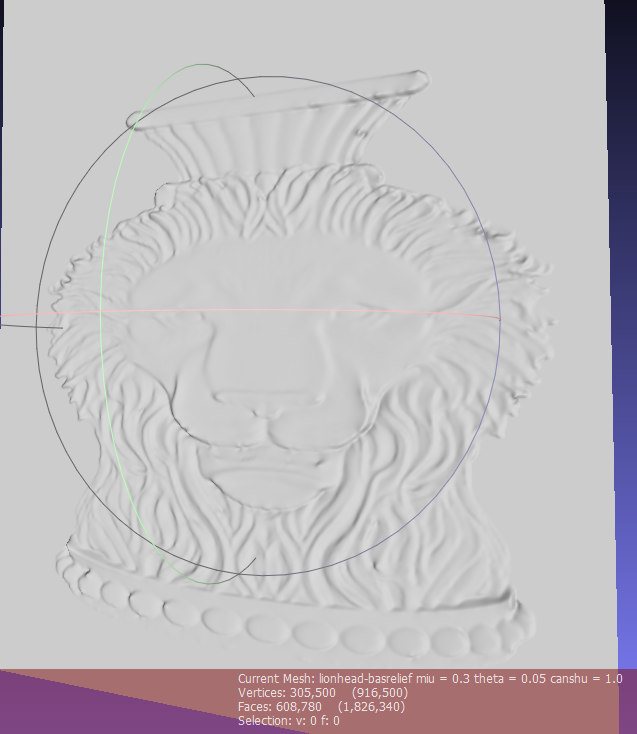
Miu is the parameter to influence strength level of height, when miu is decreasing, the more strong the height difference is. Theta is the threshold of all height, users can adjust the whole height of relief by modify that value. And last but not least, phi is the coefficient of LN, when phi > 1.0, it make the feature become stronger, so it can be reached to go further generate high feature bas-relief by modify phi, even if the input 3D model has low features.

Here is the final matrix calculation:



It looks simple, however, it a complex matrix calculation, matrix A is Laplacian template, like {-8,1,1,1,1,1,1,1,1} or {-4,1,0,1,0,1,0,1,0} or some other templates. X is the height of all pixels, it’s the value we want to calculate. And b is the divergence’s Laplacian equation, it’s generate from the normal map.

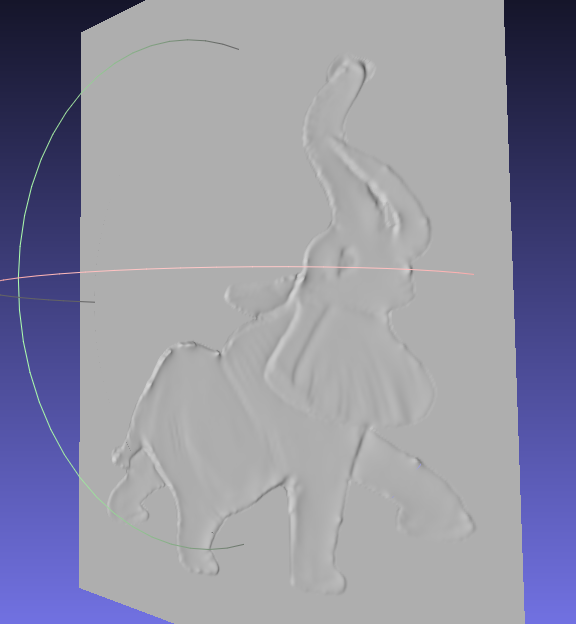
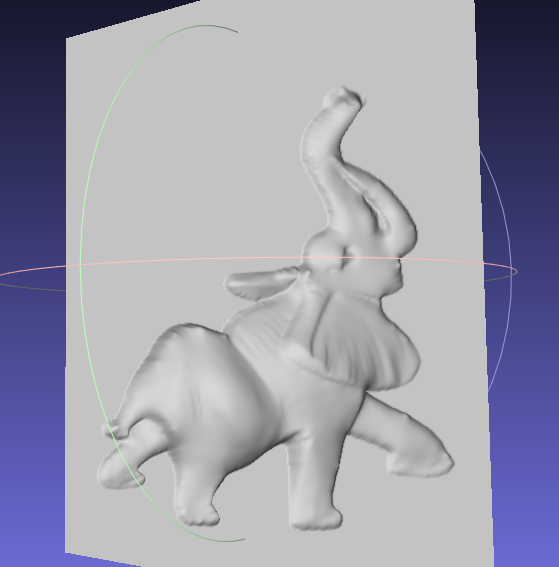
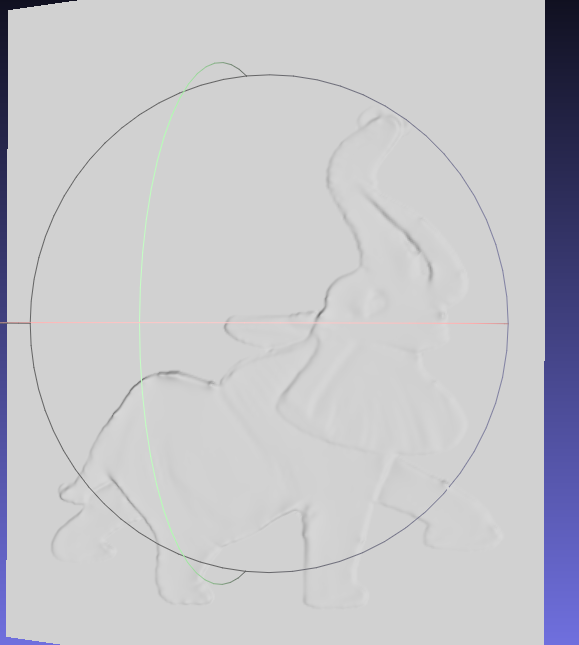
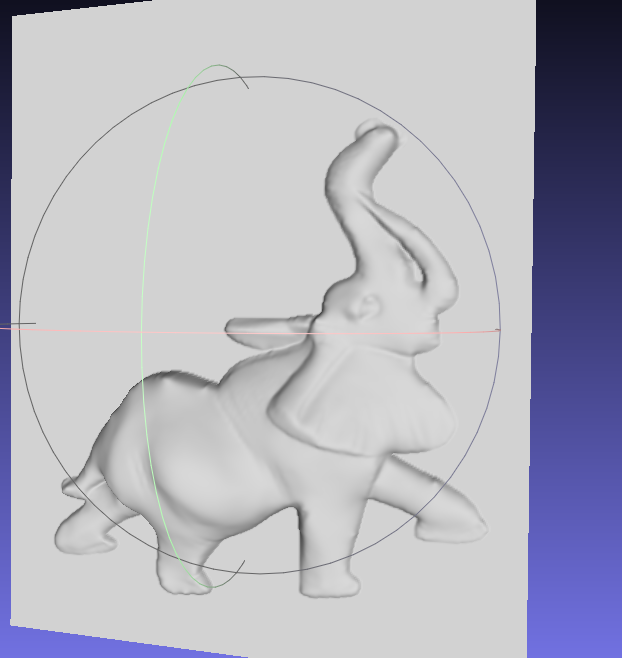
**Bas-Relief Generation and Stylization**

**(a) (b) (c)**

**Figure 6 Three different kind of bas-relief generated from the same 3D model. From left to right the parameters are (a) miu = 0.001, theta = 0.5, phi = 2.0, (b) miu = 0.3, theta = 0.05, phi = 1.0, (c) miu = 0.4, theta = 0.5, phi = 1.0.**

What I post here are three bas-relief generated from my code. It can not be denied that the more resolution it is, the more time it consumes. Trible of them has the resolution of 611 \* 500 pixels. For (a) it has lower miu value, it’s obvious this bas-relief has large difference among each pixel’s height. (b) and (c) has pretty similar miu value, so the difference between them is small but it also can be noticed.



1. (b) (c) (d)

**Figure 7 Here are four bas-relief of elephant 3D input. (a) and (c) are higher height elephant and (b) and (d) are lower height elephant.**

What’s more, convex or concave bas-relief can be generated by modify the normal vector’s second derivative calculation style. Figure 7 show one pair (a) and (b) is convex elephant with positive height dominant and the other pair (c) and (d) is concave elephant with negative height dominant.

**Process Time**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Parameter value | Valid pixel | Data treatment time | Height calculation time |
| Lion-head | Miu=0.001,theta=0.5,phi=2.0 | 91313 | 183852 | 847812 |
| Lion-head | Miu=0.5,theta=0.5,phi=1.0 | 91313 | 190786 | 223728 |
| elephant | Miu=0.001,theta=0.5,phi=2.0 | 48955 | 83735 | 327498 |
| elephant | Miu=0.5,theta=0.5,phi=1.0 | 48955 | 85129 | 97404 |

**Table 1 Four of these bas-relief based on same resolution 500\*400. From the data record, the more obvious the height is , the more calculation time it needs.**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Valid pixel | Date treatment time | Height calculation time |
| 611\*500 | 139577 | 653130 | 1.74929e+06 |
| 500\*400 | 91313 | 183852 | 847812 |
| 350\*300 | 47897 | 216429 | 331917 |

**Table 2 Same input, different resolution. It use lion-head as input with parameter miu=0.001, theta=0.5, phi=2.0. From these data, we can draw the conclusion the matrix calculation time consuming increasing faster than the resolution.**

**Conclusion and Future Work**

Following this method, a height map representing a bas-relief is constructed from a normal image and compressed nonlinearly while preserving or even enhancing features of the image. (By changing the coefficient of LN). One advantage of our algorithm is freeing of depth discontinuity intrinsically.

Our future work will focus on two direction:

One is design an user interface for this software in order to let user arbitrarily cur, rotate the 3D model so that they can get desired relief. Besides, they can adjust the feature strength by a simple operation. The other is we can generate bas-relief from grayscale picture. Because grayscale reflects the positional relationship of pixels to a certain extent.

**Reference**

*[1]Zhongping Ji, Weiyin Ma, Xianfang Sun, "Bas-Relief Modeling from Normal Images with Intuitive Styles", IEEE Transactions on Visualization & Computer Graphics, vol. 20, no. , pp. 675-685, May 2014, doi:10.1109/TVCG.2013.267*

*[2]T. Weyrich, J. Deng, C. Barnes, S. Rusinkiewicz, and A. Finkelstein, “Digital Bas-Relief from 3D Scenes,” Proc. ACM SIGGRAPH ’07, p. article 32, 2007.*