Multimed Tools Appl
https://doi.org/10.1007/s11042-018-5826-7

3D sunken relief generation from a single image by feature line enhancement

Meili Wang[1,2,3] · **Liying Yang**[1] · **Tingting Li**[1] · **Shihui Guo**[4] · **Jincen Jiang**[1] · **Hongming Zhang**[1] · **Jian Chang**[5]

Received: 1 October 2017 / Revised: 19 January 2018 / Accepted: 21 February 2018

Abstract Sunken relief is an art form whereby the depicted shapes are sunk into a given flat plane with a shallow overall depth. In this paper, we propose an efficient sunken relief generation algorithm based on a single image by the technique of feature line enhancement. Our method starts from a single image. First, we smoothen the image with morphological operations such as opening and closing operations and extract the feature lines by comparing the values of adjacent pixels. Then we apply unsharp masking to sharpen the feature lines. After that, we enhance and smoothen the local information to obtain an image with less burrs and jaggies. Differential operations are applied to produce the perceptive relief-like images. Finally, we construct the sunken relief surface by triangularization which transforms two-dimensional information into a three-dimensional model. The experimental results demonstrate that our method is simple and efficient.

Keywords Sunken relief · Unsharp Masking (USM) · Triangularization

✉ Shihui Guo
guoshihui@xmu.edu.cn

[1] Northwest A&F University, Xianyang, Shaanxi, China

[2] Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture, Yangling, Shaanxi 712100, China

[3] Shaanxi Key Laboratory of Agricultural Information Perception and Intelligent Service, Yangling, Shaanxi 712100, China

[4] Xiamen University, Xiamen, China

[5] Bournemouth University, Poole, UK

Published online: 17 March 2018

1 Introduction

Relief is a special art form between drawing and sculpture that is usually carved into a base surface or created by removing the unwanted pieces of stone, wood, and metal in a front-to-back order. It is widely used in a variety of items for signs, narratives, decorations and other purposes since ancient periods. In the modern industrial production, relief also has a broad applications, such as in producing nameplates, coins, or architectural decorations. Relief can be categorized into high relief, bas-relief (low relief), and sunken relief according to its depth and space structure.

Traditional relief is created by hand which is laborious and time consuming that requires both professional skills and artistic expertise, and once the relief is generated, it cannot be easily changed. On the contrary, digital relief is flexible and easy to be edited [2, 14]. The techniques of digital relief generation can be categorized into image based and 3D model based [5, 21]. Currently, relief generation is focused on high and bas-relief generation and little attention has been paid on sunken relief generation [1, 2, 18]. A survey of computer assisted relief generation can be found in [6].

Sunken relief is a special relief form that is mainly depicted by lines and that the texts or patterns are lower than the material plane. So compared with high relief and bas-relief, sunken relief is the most space-saving relief form. Lines are fundamental elements in the art of painting and relief, and they can effectively convey both shape and material information. Lines also play an important role in human perception. In recent years, in the field of non-photorealistic rendering, depicting shapes using feature lines has become a popular topic. The methods of extracting feature lines have also been continually innovated and improved. In reliefs, feature lines are divided into several main categories, including contours, creases, and suggestive contours [17]. Those types of lines play important roles in sunken relief generation. Recently, sunken relief generation [8, 16, 24] took 3D models as inputs and extracted feature lines first. Although satisfactory results can be obtained from 3D models, the generating process is complex and inefficient. Especially, when the complexity of 3D models increases, it is not easy to get the desired sunken relief models. 2D images are easier to obtain than 3D models. However, those image based methods are mostly focused on bas-relief generation and did not consider sunken relief generation since they did not pay attention to the importance of feature lines [16]. Therefore, there remains much space to be explored to generate sunken relief from a single image.

In this paper, we propose a simple and efficient image-based sunken relief generation method based on feature line extraction from 2D images. First, image pre-processing methods are applied to smoothen the image. Then, feature lines are extracted by finding the areas with significant differences in values of adjacent pixels. Third, unsharp masking (USM) and differential operations are used to enhance features and produce a relief effect. Finally, 3D relief models are constructed by a triangulated mesh, in which pixel values are considered as depth information.

Our goal is to generate sunken relief from a single image based on some image processing methods which is simple and applicable to many kinds of input images, such as animals, architectures, and cartoons. The result of this work can be served as the input for industrial manufacturing. The contributions of this paper are as follows:

1. A novel method to extract feature lines for sunken relief generation is presented, and we show how to obtain relatively better feature lines which contain as many details as possible and with few breakpoints and noises by adjusting the threshold.
2. A triangularization method to generate the 3D sunken relief using image pixel values is implemented based on the sunken relief-like images. This process is essential for the

whole experiment, in which we can manually set the offset value and compare the corresponding results to obtain a plausible 3D shape.

The rest of the paper is organized as follows: Section 2 reviews related works on digital relief generation, feature lines extraction and the USM algorithm. Section 4 is the systematic overview of our algorithm. Section 5 gives the detailed descriptions of feature lines extraction and feature enhancement. Section 5 introduces the method to obtain 3D sunken relief and compares the results of our method and previous methods. Section 6 gives the conclusions and suggestions of future works.

2 Related works

2.1 Digital relief generation

Digital high- and bas-relief generation from 3D models has been widely investigated [1, 2, 18, 25]. In addition, different from these methods that generate relief from a certain still model, recently, Wang et al. [19] successfully selected an informative representative pose from an animation sequence and applied the pose to bas-relief generation. However, for the sunken relief, there remains much space to be explored. Most studies [8, 16, 24] took three-dimensional (3D) models as inputs and extracted feature lines first. Then, the final sunken relief was generated by engraving the feature lines into a flat plane. Although satisfactory results can be obtained from a simple 3D model, the algorithm process is complex. Especially, if the 3D model is complex, the generated sunken relief is less clear than that generated from a simple 3D model [17, 24]. Two-dimensional (2D) images are easier to obtain than 3D models. Therefore, Wang et al. [15] proposed an image-based algorithm that adopted gradient operations to convert an image into a relief, and then solved a Poisson equation to construct the depth information. Hai et al. [21] build the face parts map region (FPM-R) to detect the hair, eyes, eyebrows, nose and lips and to make them protrude for bas-relief generation. Wu et al. [16] presented a bas-relief generation approach using a 2D image of a human face. First, an image of bas-relief was generated from the input image. Then the shape-from-shading technique was applied to determine the 3D shape of the final bas-relief. Zeng et al. [22] also proposed a bas-relief generation algorithm based on a single image. They extracted feature lines first, then generated and enhanced a base surface using both intensity and gradient information. They also introduced a feedback process to prevent depth errors which arose during the enhancement process. Wu et al. [20] first obtained a rough shape of bas-relief and then detected image details. A detailed enhancement bas-relief was generated by combining the two parts. Lu et al. [9] proposed a hybrid method to generate bas-relief of human faces using both 3D depth images and 2D intensity images. Miao et al. [10] proposed a novel sculpturing technique to generate some complex sculptures which carved the input drawing lines into a 3D model. Sohn [13] proposed a new method to automatically generate digital bas-reliefs with input images and depth map even on smart phones. All these mentioned methods generated 3D bas-relief based on an image and obtained satisfactory results. To investigate the sunken reliefs, it can be seen that lines are important features in depicting the scene or story. Basically, they are divided into three main categories, including contours, creases and suggestive contours [17]. However, although existed methods could obtain bas-relief they did not pay attention to the importance of feature lines in sunken relief generation [16] for that the shape features of sunken reliefs depicted by feature lines are obvious. Therefore, there remains much space to be explored to generate sunken relief from a single image. This paper aims to generate sunken relief from a single image based on some image processing methods which is simple and applicable to many kinds of input images with high efficiency.

2.2 Feature lines extraction

As a form of sculpture, sunken relief is mainly generated by carving the lines into a smooth plane. Most studies focus on adopting complex algorithms to generate smooth curved surfaces whose depth varies within a limited range. However, if only the feature lines are engraved into the plane, we can easily generate a sunken relief. Despite this, few researchers were aware of the importance of feature lines for relief sculptures until Wang et al. [16] proposed an innovative method based on line-drawings. On the basis of this study, Wang et al. [17] and Zhang et al. [24] further investigated line drawings and relief generation from a 3D mesh.

Methods to extract feature lines can be roughly classified into two categories: object-space and image-space approaches [7, 23]. Object-space algorithms extract feature lines directly on 3D surfaces by seeking out points whose radial curvature is zero. Such approaches are more complex than image-space algorithms, which extract feature lines from images by image processing after rendering [3, 12].

2.3 USM

Owing to the limited dynamic range, the influence of the light, and the image device restrictions, the quality of the image will be degraded during the acquisition process. Therefore a lot information in the original image cannot be recognized by human eyes. Image sharpening is an image enhancement method that uses various mathematical methods and transformations to improve image contrast and sharpness, to highlight details in the image. USM originates from traditional photographic technology. It is an image edge sharpening algorithm which is based on image convolution. The principle of USM is to exaggerate the lighter-darker contrast between the two sides of an edge for the sake of enhancing the visual definition of the image [4, 11]. The principle of the classic linear USM is that first the original image is smoothed by a linear high-pass filter and then multiplied by a scale factor. The result of this is added to the original image to obtain the enhanced image. In this paper, we mainly adopt USM to enhance the feature lines with some local operations to facilitate the final image for sunken relief generation.

3 Method overview

First, the matrix of the 2D image is processed by image processing techniques such as morphological expansion, opening and closing operations and mean filtering. Then we extract the feature lines and remove splashes in the feature line image. After that, USM technology is used to sharpen and enhance the contours and detail information. We define the feature lines as those that are engraved to enhance the local details. Processing is necessary to smooth the feature lines. The sunken relief image is generated by a differential operation. Finally, the 3D relief is generated by triangularization. The framework is shown in Fig. 1.

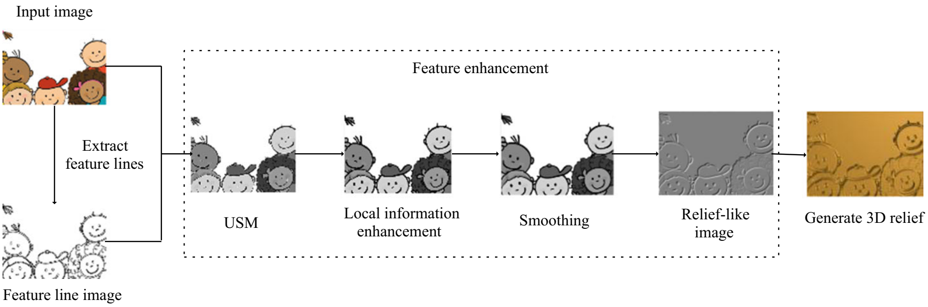


Fig. 1 Framework of our algorithm

4 Feature line extraction and feature enhancement

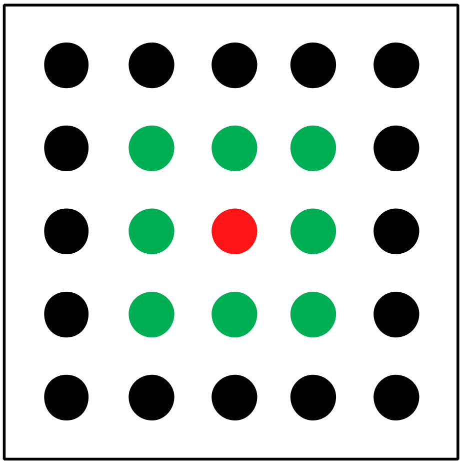
4.1 Feature line extraction

Feature lines are based on the 2D source image. As with any edge-detection algorithm that operates in a continuous domain, a threshold parameter is necessary to adjust quality of the results. Thus, we extract feature lines from the original image through threshold detection.

The general idea is to compare the pixel value of the currently selected point with pixel values of adjacent points. If a difference above the threshold value exists, we consider that this selected point belongs to the boundary of the area; otherwise, the point does not. As shown in Fig. 2, the red point is defined as the selected point and the eight green points are the adjacent points. If the difference between the pixel value of the red point and that of any of the green points is greater than the pre-set threshold, this point is extracted as part of a feature line.

To simplify the program code, considering that boundary pixels of the image have little effect on the final results of the feature line image, we neglect all the points lying on the image border; that is, the first and last rows and the first and last columns of the gray matrix are ignored when scanning the whole image row by row and column by column. Thus, we can ensure that there are always 8-neighbors for every test point and fewer constraint conditions need to be considered.

Fig. 2 Threshold detection



The algorithm of threshold detection for feature line extraction is as follows. Convert RGB values of the input image to grayscale values by forming a weighted sum of R, G, and B components. Obtain the gray matrix $I(i,j)$ where i represents a row and j represents a column of the matrix.

A transformation S is performed on $I(i,j)$ with the purpose of conveniently determining the range of differences among the pixel values and easily detecting the target point. In general, a sine function is selected with a range of [0, 1] multiplied by a constant m so that the range is changed to $[0, m]$. The corresponding transformation is as follows:

$$S(I(i,j)) = m^{*}\sin(I(i,j)) \tag{1}$$

Set t as the threshold. Scan the image row by row, and compare the selected point with its 8-neighbors. If there exists one neighbor pixel with its pixel value difference bigger than the threshold, this point will be set to black (the pixel value is zero) and selected as part of the feature lines. Otherwise, it is not on the feature lines and will be set to white (the pixel value is 255). The condition is as follows:

$$I(i,j){-}I(a,b) > t \tag{2}$$

where a represents the line number of neighbors which can be taken as $i-1$ and $i+1$, and b represents the column number of the neighbors which can be taken as $j-1$ and $j+1$.

The threshold is of great importance for the quality of the feature line image. When t is small, the extracted feature lines convey many details. As the value of t increases, the feature line image contains less details and line fractures increase. However, the line becomes thinner and burrs decrease. Therefore a suitable threshold results in a feature line image with many details but few line fractures. Repeated experiments we have performed show that good results can be achieved when the threshold is in the range of 0.3 to 2.5. Figure 3 shows feature line images for three different thresholds. A threshold of 0.25 results in a detailed image but with many burrs, whereas a threshold of 3.0 results in an image with many fractures.

Although a suitable threshold can improve the quality of feature line image, there still exists noise or blurriness in image; therefore, image processing is necessary before extracting feature lines. We smoothen the image using morphological opening and closing operations, and then apply a median filter. After extracting the feature lines, there are still some false edges in the feature line image affecting the accuracy of the extraction. Therefore, we remove these points by setting the pixel value to 255 when the point's pixel value is zero and all its 8-neighbors greater than zero. The condition is as follows:

$$I(i,j) = 0\&\&I(a,b) > 0 \tag{3}$$

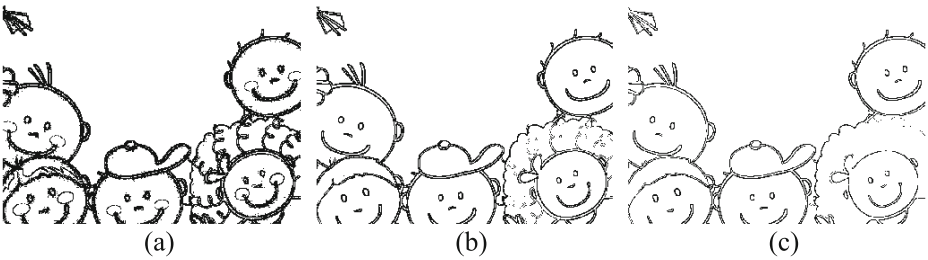


Fig. 3 Feature line images created by our algorithm using thresholds of (**a**) 0.25, (**b**) 1.1, and (**c**) 3.0

As a result, an image composed of black and white lines is obtained, effectively extracting the feature lines in the original image.

4.2 Feature enhancement

After obtaining the feature lines image, we can directly generate a relief-like image; however, it contains a lot of line fractures (see Fig. 4). To overcome this, we apply some image processing to the input image to sharpen the feature lines and to enhance the final relief quality (see Fig. 8a).

Unsharp masking USM is a commonly used technique which is used to sharpen the edges of an image. By using USM, the contrast of the edge details can be quickly adjusted. A bright line and a dark line are generated on the two sides of any edges to make the image more distinct. In this paper, we used the classic linear USM because it is simple and the enhancement effect is relatively good.

First the image is smoothed. We apply the neighborhood average method which adds the gray value of one pixel in the original image to the gray value of the pixel adjacent to it and then the average is calculated by dividing by 16. In this process, a template is needed. We defined a Gaussian template denoted by W which is a 3 × 3 matrix. The template can be seen as follows:

$$W = \begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix} \tag{4}$$

To obtain the average in the next step, we unitize it as follows:

$$W_u = \frac{1}{16} * W \tag{5}$$

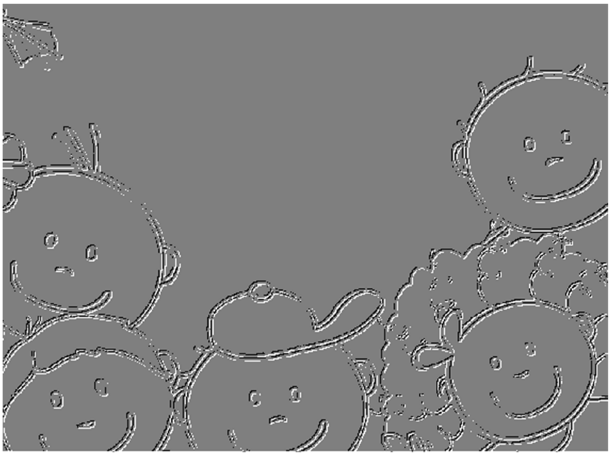


Fig. 4 Relief images with breaks

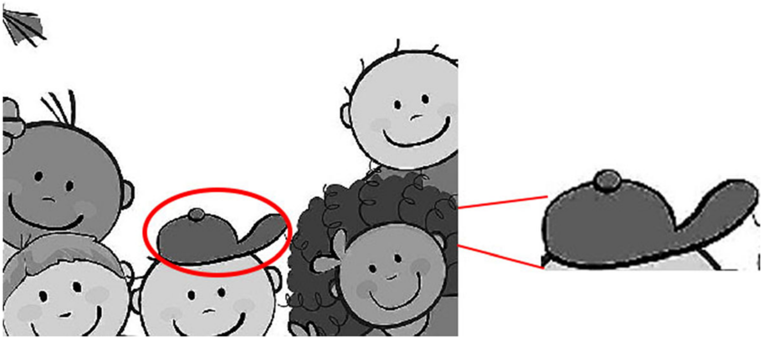


Fig. 5 Results of the unsharp masking

Every pixel in the original image is multiplied by the corresponding value of the template. The process is as follows:

$$\begin{aligned} g(i,j) = {} & W_u(1,1)*f(i-1,j-1) + W_u(1,2)*f(i-1,j) + W_u(1,3)*f(i-1,j+i)+ \\ & W_u(2,1)*f(i,j-1) + W_u(2,2)*f(i,j) + W_u(2,3)*f(i,j+1)+ \\ & W_u(3,1)*f(i+1,j-1) + W_u(3,2)*f(i+1,j) + W_u(3,3)*f(i+1,j+1) \end{aligned} \tag{6}$$

where $f(\cdot,\cdot)$ is the gray value of the original image.

Since the smoothed image is the low frequency part of the image, to obtain the high frequency part, we subtract it from the original. Then the high frequency part is multiplied by a factor and added back to the original. This process is expressed as follows:

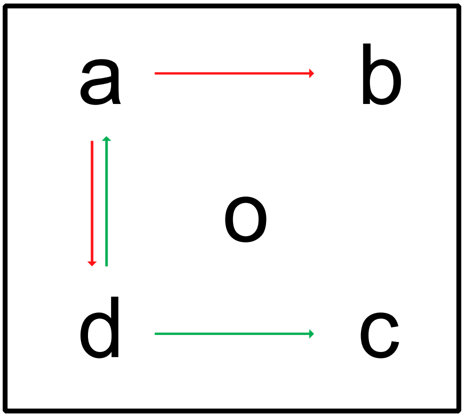
$$G(i,j) = f(i,j) + k(f(i,j)-g(i,j)) \tag{7}$$

where $g(i,j)$ is the smoothed version which obtained by Eq. 6, and k is the factor representing the amount of enhancement. We set $k = 5$ according to experiments.

The image after USM is seen in Fig. 5, and we can see from the enlarged hat that the borders of objects in the image are sharper.

Local information enhancement and smoothing After USM, the border is sharper, but the obtained image quality is poor (see Fig. 5). Therefore, we apply local information

Fig. 6 Local information enhancement



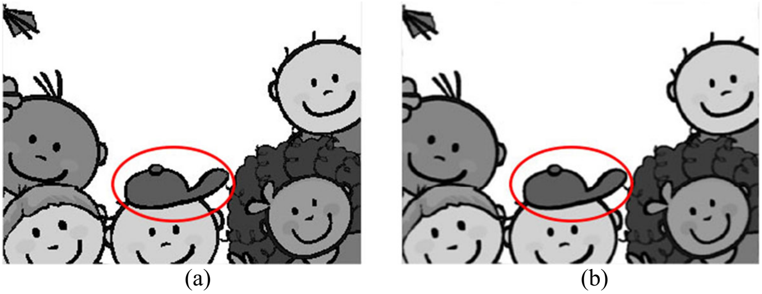


Fig. 7 Images showing (**a**) the result of local information enhancement in which a black line warps the contour line and (**b**) the result of Gaussian smoothing with less burrs and jaggies than (**a**)

enhancement processing for a better effect. In this process, we define the points whose pixel value is 0 as the domain of definition to process the original image because pixel value of the point in the feature line is 0.

In principle, the original image is scanned row by row and changes are made to points on the feature lines if there are differences between that point and its neighbors. We set the pixel value difference to 64 because there can be a distinction generally when it is 64. For example, in Fig. 6, supposing o is the point on the feature line, we first traverse points of the image in directions of $\overrightarrow{ab}$ and $\overrightarrow{ad}$. If there is a case that $a < c$ and $|a - c| \geq 64$ (i.e., a is darker than c), we modify it to satisfy $|a - c| < 64$ by increasing the value of a and reducing the value of c. Then we set o to zero. The same work is done in directions of $\overrightarrow{da}$ and $\overrightarrow{dc}$.

The result is that a black line is added to lighter parts. As shown in Fig. 7a, a black line warps the contour line and through the differential operation, this black line will be transformed into a sunken curve. But the black line is rough, so we use Gaussian smoothing to smooth it. As seen in Fig. 7b, burrs and jaggies are reduced.

Differential operation Generating the sunken effect is a key step for digital sunken relief generation. Considering the image processing, we can obtain the relief effect from it. The convex or concave effect showed in many images is obtained by implementing a differential operation. A differential operation is a process in which the present value is subtracted from the next value

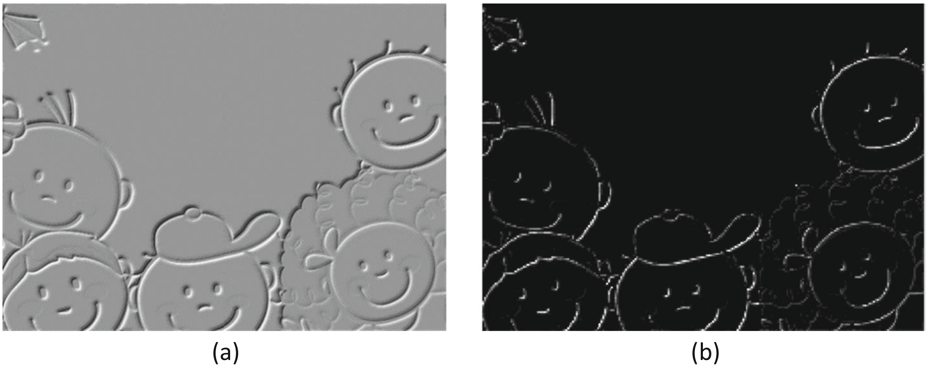
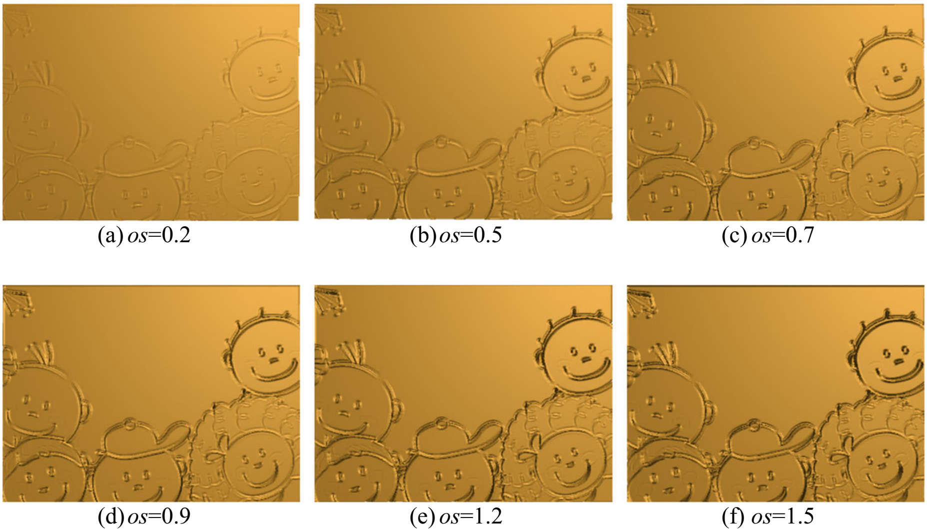


Fig. 8 Sunken relief-like image of children. (**a**) Generated through Eq. 10 and (**b**) generated after deleting the constant 0.5 in Eq. 10


(a) *os*=0.2 (b) *os*=0.5 (c) *os*=0.7

(d) *os*=0.9 (e) *os*=1.2 (f) *os*=1.5

Fig. 9 Different sunken reliefs when *os* takes different values

(forward difference, see Eq. 8) or the previous value is subtracted from the present value (backward difference, see Eq. 9). These two differential operations can generate a concave effect.

$$\Delta f(x) = f(x+1) - f(x) \tag{8}$$

$$\nabla f(x) = f(x) - f(x-1) \tag{9}$$

In this paper, we apply the differential operation to obtain the perceptive sunken relief. We use a linear spatial filtering function, whose principle is convolution, to achieve the differential operation. However, for most images, low-frequency components often occupy the dominant position; that is, the images are based on low frequency components. Therefore, most results are small or even zero and thus the overall color tend to be black. To obtain better visual effects that the color is close to

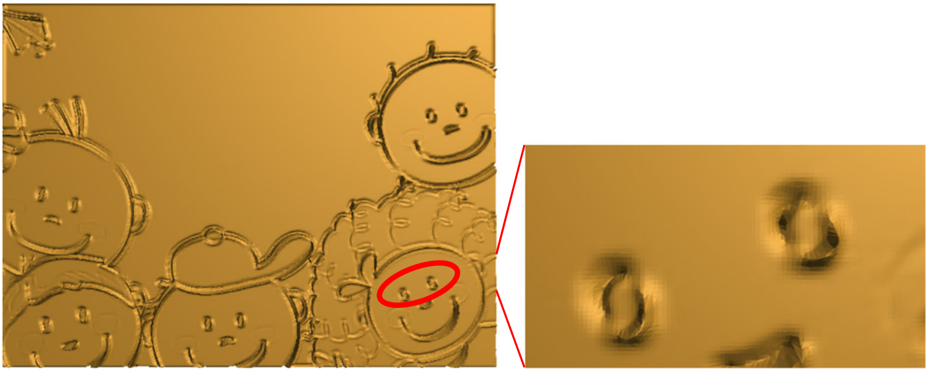


Fig. 10 Sunken relief and enlarged eyes when *os* takes the value of 1.2

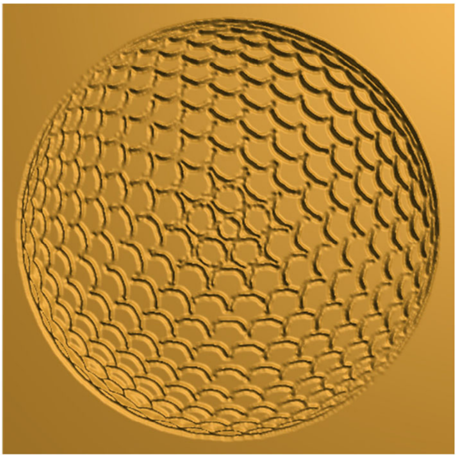


Fig. 11 Sunken reliefs generated by our method

the lime color, we add a direct component in such a result, that is, to increase a pixel value constant to ensure a certain gray level. The process is as follows:

$$F(i,j) = 0.5 + \sum_{k,l} G(i{-}k, j{-}l)h(k,l) \tag{10}$$

where $h(k,l)$ is the convolution kernel defined as $h = 1 \quad 0 \quad 0{-}1$, k and l represent rows and columns of $h(k,l)$ respectively and the constant 0.5 is the direct component. The constant can enhance the brightness of the image and will contribute to the 3D sunken relief. According to our experiments, when the constant is 0.5, the result is relatively better. The result is shown in Fig. 8.

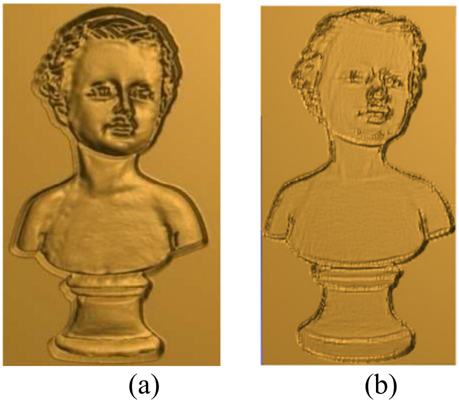
5 Sunken relief generation and comparisons

5.1 Triangularization

A triangulated mesh is adopted to construct 3D relief models after obtaining the required image information from the 2D image processing. For a simple implementation, the i and j

Table 1 Time cost of our algorithm for three different images

Name (faces)	Time cost (seconds)				
	Feature line extraction	Feature enhancement	Triangularization	Total	Average time for each 10,000 faces
Bust (557106)	0.089	0.703	13.560	14.352	0.253
Children (804420)	0.089	0.537	19.613	20.239	
Horse (839808)	0.114	0.533	20.419	21.066	



(a) (b)

Fig. 12 Sunken relief of a bust by (**a**) Wang et al. [17] and (**b**) our method

components of each vertex position correspond to the location of their counterpart in the line image *F*. Accordingly, connectives of 3D vertexes at each pixel constitute the triangular mesh.

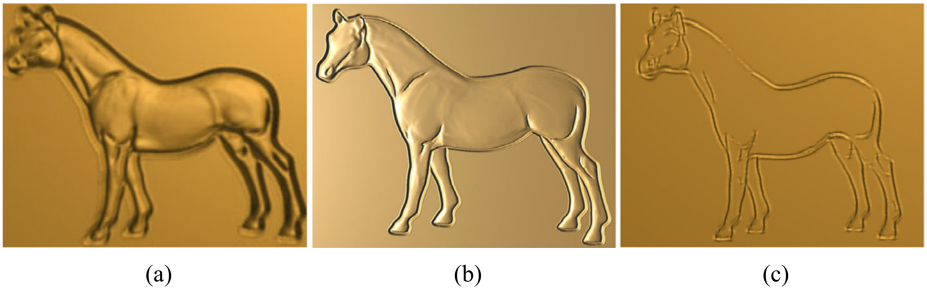
$$z = F(i,j) - os \tag{11}$$

where $F(i,j)$ is the image pixel value, which corresponds to depth z, os is the offset value. This leads to a sculpture in which the background is mapped to a zero-level and each line is carved deeper into the material.

Generally, the engraving depth is different when the parameter os takes different values. Repeated experiments show that the larger os is, the deeper feature lines will be engraved. However, when os increases to a certain value, the engraved line becomes unnatural. Figure 9 shows different results when os is 0.2, 0.5, 0.7, 0.9, 1.2, and 1.5 respectively. We can see that when os takes the value of 0.9, the sunken relief is most lifelike in appearance; when os is smaller than 0.9, carved lines are relatively shallow; when os takes the value of 1.2 or larger, there have been some abnormal deformations, especially lines of children eyes (see Fig. 10). In addition, Fig. 11 shows the sunken relief obtained by our algorithm.

5.2 Software of comparisons

Our algorithm mainly uses image processing to construct a 3D sunken relief from a 2D image. We implemented the algorithm in Matlab. All experiments were tested on a 3.20 GHz Intel CPU with 8 GB RAM assisted by a NVIDIA GeForce GTX 750 graphics card.



(a) (b) (c)

Fig. 13 Sunken relief of a horse by (**a**) Wang et al. [17], (**b**) Zhang et al. [24] and (**c**) our method

Compared with the object-based method, one important advantage of our method is that it does not require costly computation and can be easily implemented in graphics hardware. Table 1 shows the computation time of our algorithm for four images. From Table 1, we can see that our method is more efficient.

Experimental results verify that the proposed method is effective for generating a sunken relief from a single 2D image. For complex images, our method can better maintain detail information. Figures 12 and 13 compare results produced by our method with those produced using the method of Wang et al. [17] and Zhang et al. [24]. Compared with Wang's and Zhang's methods, our method can obtain a lifelike effect and is less time-consuming. Any image can be processed to generate 3D sunken relief; therefore, development of industrial production for sunken relief can be greatly promoted. However, for images with intensive lines, our method needs to be further improved.

6 Conclusions and future works

In this paper, we proposed a simple and effective method to generate sunken relief from a single image focused on feature lines enhancement. We adopted some image pre-processing to smooth the original image and improve the quality of the image obtained by feature line extraction. Local information enhancement and differential operation were applied to enhance feature information. We obtained a smooth and distinct relief-like image. Finally, a triangulated mesh was applied to construct 3D relief models through triangularization. Experiments showed that results were authentic and vivid. However, all lines of the generated sunken relief have the same engraving depth and the height transition among lines has not been considered. Therefore, we will concentrate on carving lines in different depth which will better convey the layer of the sunken relief in our future work. Another important direction is to explore designing and crafting stylized sunken relief which will make sunken relief more natural and vivid.

Acknowledgments This work was funded by the National Natural Science Foundation of China (61402374, 61702433, 61661146002). We thank all reviewers for editing the English of this manuscript.

References

1. Arpa S, Süsstrunk S, Hersch R (2015) High relief from 3D scenes. Comput Graphics Forum 34(2):253–263
2. Cignoni P, Montani C, Scopigno R (1997) Computer-assisted generation of bas-and high-reliefs. J Graph Tools 2(3):15–28
3. Decarlo D, Finkelstein A, Rusinkiewicz S, Santella A (2003) Suggestive contours for conveying shape. ACM Trans Graph 22(3):848–855
4. Deng G (2011) A generalized unsharp masking algorithm. IEEE Trans Image Process 20(5):1249–1261
5. Hai TT, Sohn BS (2017) Bas-relief generation from face photograph based on facial feature enhancement. Multimed Tools Appl 76(8):10407–10423
6. Kerber J, Wang M, Chang J, Zhang J, Belyaev A, Seidel H (2012) Computer assisted relief generation-a survey. Comput Graphics Forum 31(8):2363–2377
7. Lee Y, Markosian L, Lee S, Hughes J (2007) Line drawings via abstracted shading. ACM Trans Graph 26(3):18

8. Liu S, Xu X, LI, B., Zhang, L. (2011) An algorithm for generating line-engraving relief. J Chin Comput Syst 32(10):2088–2091
9. Lu Q, Wang L, Meng X, Wang W (2015) The bas-relief generation method of human faces from 3D depth images and 2D intensity images. J Comput-Aided Des Comput Graphics 27(7):1172–1181
10. Miao Y, Chen M, Fang X (2016) 3D model sculpturing technique based on drawing lines. J Comput-Aided Des Comput Graphics 28(1):50–57
11. Ramponi G (1998) A cubic unsharp masking technique for contrast enhancement. Signal Process 67(2): 211–222
12. Raskar R (2001) Hardware support for non-photorealistic rendering. In: ACM SIGGRAPH, eurographics Workshop on Graphics Hardware, Association for Computing Machinery, pp 41–46
13. Sohn B (2017) Ubiquitous creation of bas-relief surfaces with depth-of-field effects using smartphones. Sensors 17(3):572
14. Song, W., Belyaev, A., Seidel, H.P. (2007) Automatic generation of bas-reliefs from 3D shapes. In: IEEE International Conference on Shape Modeling and Applications, IEEE Computer Society, pp 211–214
15. Wang M, Chang J, Pan J, Zhang J (2010) Image-based bas-relief generation with gradient operation. In: Proceedings of the 11th IASTED International Conference Computer Graphics and Imaging, Acta Press, Innsbruck, pp 33–38
16. Wang M, Kerber J, Chang J, Zhang J (2011) Relief stylization from 3D models using featured lines. In: Spring Conference on Computer Graphics, ACM, pp 37–42
17. Wang M, Chang J, Kerber J, Zhang J (2012) A framework for digital sunken relief generation based on 3D geometric models. Vis Comput 28(11):1127–1137
18. Wang M, Sun Y, Zhang H, Qian K, Chang J, He D (2016) Digital relief generation from 3D models. Chinese J Mech Eng 29(6):1128–1133
19. Wang M, Guo S, Liao M, He D, Chang J, Zhang J, Zhang Z (2017) Pose selection for animated scenes and a case study of bas-relief generation. In: Computer Graphics International Conference, ACM, pp 31
20. Wu W, Liu L (2016) Course-to-fine bas-relief generation algorithm from images. College Mathematics 32(6):1–7
21. Wu J, Martin R, Rosin P, Sun X, Langbein F, Lai Y, Marshall A, Liu Y (2013) Making bas-reliefs from photographs of human faces. Comput Aided Des 45(3):671–682
22. Zeng Q, Martin R, Wang L, Quinn J, Sun Y, Tu C (2014) Region-based bas-relief generation from a single image. Graph Model 76(3):140–151
23. Zhang L, He Y, Xie X, Chen W (2009) Laplacian lines for real-time shape illustration. In: 2009 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, Association for Computing Machinery, Boston, pp 129–136
24. Zhang Y, Zhou Y, Li X, Zhang L (2013) Line-based sunken relief generation from a 3D mesh. Graph Model 75(6):297–304
25. Zhang Y, Zhang C, Wang W, Chen Y (2016) Adaptive bas-relief generation from 3D object under illumination. Comput Graphics Forum 35(7):311–321

Meili Wang is an associate professor at College of Information Engineering, Northwest A&F University. She received her PhD degree in computer animation in 2011 at the National Centre for Computer Animation, Bournemouth University. Her research interests include computer graphics, geometric modelling, image processing, visualization and virtual reality.