Feature Sensitive Bas Relief Generation

Jens Kerber[1], Art Tevs[1], Alexander Belyaev[2], Rhaleb Zayer[3], Hans-Peter Seidel[1]
[1]MPI Informatik, Saarbrücken, Germany
[2]Heriot-Watt University, Edinburgh, Scotland, UK
[3]INRIA, Nancy, France

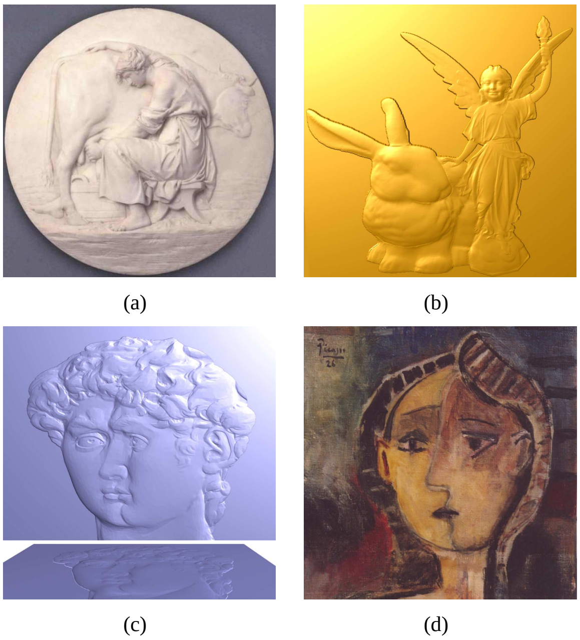


Fig. 1. (a) a real world bas relief; (b) a geometric montage of 3 models; (c) a cubism like relief of the David head model; (d) an original Picasso painting

***Abstract*—Among all forms of sculpture, bas-relief is arguably the closest to painting. Although inherently a two dimensional sculpture, a bas-relief suggests a visual spatial extension of the scene in depth through the combination of composition, perspective, and shading. Most recently, there have been significant results on digital bas-relief generation but many of the existing techniques may wash out high level surface detail during the compression process. The primary goal of this work is to address the problem of fine features by tailoring a filtering technique that achieves good compression without compromising the quality of surface details. As a secondary application we explore the generation of artistic relief which mimic cubism in painting and we show how it could be used for generating Picasso like portraits.**

***Keywords*—shape deformation, computer art, sculpture, tone mapping**

1. Introduction

Among all forms of sculpture, bas-relief is arguably the closest to painting. Although inherently a two dimensional sculpture, a bas-relief suggests a visual spatial extension of the scene in depth through the combination of composition, perspective, and shading. This suggestive power of bas-relief has been recognized since antiquity and most of civilizations used it as a form of decoration. Nowadays bas relief remains indispensable to several modern applications e.g. coinage and packaging. It continues to thrive into the digital age where it is suitable for virtual shape decoration and computer art.

In recent research on automatic bas-relief generation from 3D scenes, the key ingredient is the so called height field (also known as depth map or range image) which is a 2.5D descriptions of the scene that encodes distance information on a 2D regular grid $z = I(x, y)$. Unfortunately the height information cannot be used directly for bas relief generation and a compression step is often required. Many of the existing bas relief generation techniques differ mainly in the compression step and as a general remark they remain limited to single objects or very simple scenes. Additionally, the fine features of surfaces often get lost throughout the compression process.

The primary goal of this work is to address the problem of fine features by tailoring a filtering technique that achieves good compression without compromising the quality of surface details. On the application side we investigate the impact of perspective on the bas relief generation. For this purpose we study the problem of merging multiple perspectives into one single scene, a problem better known in painting as cubism. The results of our approach are illustrated by generating Picasso like portraits.

2. Related Work

A bas-relief generally exhibits a negligible spatial extent, but when contemplated from an almost orthogonal vantage point it appears like a full 3D scene. The study of this phenomenon in [1] suggests that under certain transformations shading and shadowing remain unchanged in comparison to the initial scene, as long as the perspective does not change significantly.

The generation of bas-relief from virtual scenes was first studied in the pioneering work of [2]. In order to infer depth, they rely on height fields along with a perspective foreshortening in the sense that objects closer to the viewer are kept salient, whereas those in the background are mapped to a smaller z-range. Within this framework, the desired compression ratio is achieved through a linear re-scaling. Although many of the major aspects of bas-relief generation have been addressed in this framework, the method is not of general use as the visibility of small object features suffers to a great extent when linear re-scaling is applied.

Most recently, bas-relief generation has regained interest in the graphics community and there has been an increasing effort to address some of the challenges mentioned above [3], [4], [5]. This development finds inspiration in high dynamic range compression (henceforth HDRC). The purpose of HDRC is the compression of a large luminance interval of a high dynamic range image (HDRI) in such a way that it can be displayed on regular monitors without losing visually important features [6].

Bas-relief generation can be regarded as a geometrical analogue of tone mapping. Instead of the luminance interval length it focuses on the size of the depth interval and aims on producing a flat representation of the shape by keeping significant details.

The method proposed in [3] relies on the combination of a saliency measure and a feature enhancement technique. As the processing is performed in differential coordinates the bas relief is reconstructed as the solution to a diffusion equation. The approaches by [4] and [5] operate in the gradient domain. They artificially enhance certain frequencies of the gradient images in order to better preserve their visibility in the compressed outcome. Both techniques can be regarded as variants of the gradient domain high dynamic range compression algorithm proposed by [7]. A discussion and a comparison of the above mentioned bas-relief generation methods can be found in [8]. The most recent work in this young research area is [9]. The authors apply a modification of an image contrast enhancement technique. The difference to the other approaches is that it operates immediately on the height field and uses gradient information only additionaly.

As geometric features are quite different from those found in images, the adaption of HDRC techniques to shapes is far from being straightforward. In this paper we introduce a modified version of the tone mapping approach initially presented in [10], which uses bilateral filtering for decomposing an image into a base layer and a detail layer. This technique is initially targeted at reducing the contrast of a HDR images without losing intensity steps.

Cubism is a direction in painting which aims at covering multiple perspectives in a single canvas. These perspectives partly overlap and the transition areas are blended such that a continuous impression comes up. The most famous representatives of this genre are Picasso's portraits or Escher's landscapes. In [11] a technique for non photo realistic rendering is presented which allows combining multiple simultaneous viewpoints in a single image. The author illustrates how this technique could be used for storytelling and infering motion. In [12] the authors describe a method which uses general linear cameras (GLCs), recently studied and compared in [13], and a blending technique. The user is required to align the different renderings and the algorithm generates seamless results.

Our bas-reliefs can serve as input for further applications in computer art e.g. virtual carving embossment or engraving. They can serve as input for a semi automatic approach [14] or assist an artist in producing a virtual piece of art by a completely interactive tool [15]. Moreover our results can be applied as displacement maps for virtual shape decoration [16].

3. Algorithm Description

3.1 Overview

In this section we shall describe our bas relief generation approach. The input for our algorithm could be an already generated height field or a full 3D scene. In the latter case, a height field could be obtained by reading the z-buffer after an orthographic or a perspective projection.

The resulting depth maps generally are not of practical use for shape decoration as in general they exceed the range of available material depth and therefore they need to be flattened. The delicate task in bas relief generation is to devise suitable height filed compression without sacrificing the visual perception of important features.

Our method operates in the gradient domain and makes use of different binary masks in order to identify pixels which belong to sensitive parts of the height field. The gradients are decomposed into a coarse and a detail level using the bilateral filter described in [10]. The detail part is then enhanced relative to the coarse components in such a way that sensitive features will remain perceivable in the result. The new gradient images are then reassembled to obtain the final bas-relief. The user can either specify the desired compression ratio or the maximal allowed value range for the bas-relief.

Our approach capitalizes on a fundamental property of bilateral filtering. In image processing, this type of filtering is known for its edge preserving capabilities. As we are operating in the gradient domain this property translates to ridge preservation, as ridges are naturally the edges of the gradient field. In other words, the filtering preserves curvature extrema, which contain the most important information about the structure of the shape. Gaussian smoothing on the other hand as it is used for gradient frequency decomposition in [4] may wash out those ridges.

3.2 Preprocessing

Let $I(x, y)$ be the input range image. It consists of foreground pixels which describe the distance of scene objects to the camera and a background area that is filled with a certain default value δ. We first extract a binary mask $B(x, y)$ by labeling background pixels as zeros and the foreground pixels as ones.

$$B(i,j) = \begin{cases} 0, & \text{if } I(i,j) = \delta \\ 1, & else \end{cases} \tag{1}$$

We normalize the input image in a way that the smallest foreground value is mapped to the background level, such that the interval ranges form 0 to a certain value.

$$\hat{I}(x,y) = B(i,j) \cdot (I(x,y) - I_{min}) \tag{2}$$

Where I_{min} indicates the smallest foreground value of I. This helps establishing the initial value range.

3.3 Relief Generation

After the preprocessing, we compute the gradient images I_x and I_y of $\hat{I}$ by a differential quotient in each dimension. As we are in a discrete setting the formulas read as simple as:

$$\hat{I}_x(i,j) \approx \hat{I}(i+1,j) - \hat{I}(i,j) \tag{3}$$

$$\hat{I}_y(i,j) \approx \hat{I}(i,j+1) - \hat{I}(i,j) \tag{4}$$

Since background values are usually very different from the foreground values we end up with rather large gradients along the outlines of scene objects. In [5], [4] this problem was addressed by introducing a user defined threshold that sets all gradients above it to 0. We note that these discontinuities occur only along the objects' silhouette. Therefore, we can detect this area automatically with the help of the background mask gradients.

$$\begin{aligned} B_x(i,j) &\approx B(i+1,j) - B(i,j) \\ B_y(i,j) &\approx B(i,j+1) - B(i,j) \\ S(i,j) &= \begin{cases} 0, & \text{if } |B_x(i,j)| = 1 \text{ or } |B_y(i,j)| = 1 \\ 1, & else \end{cases} \end{aligned}$$

Here, S represents a binary mask that determines the boundary region and we can simply erase the silhouette pixels in the gradient images. In order to keep this exposition concise we use the notation $k \in \{x,y\}$ from now on.

$$I'_k = S \odot I_k \tag{5}$$

Where, the $\odot$ operator indicates componentwise multiplication.

In general, the resulting shape also exhibits large jumps on its surface which would negatively affect the quality of the result if they were preserved. On the one hand, they would keep the depth interval size artificially high and on the other hand large features would be too dominant in the result in a way that would drastically impair the visibility of smaller features. Therefore, we rely on an outlier detection to locate gradient entries which differ largely from the other ones.

$$O(i,j) = \begin{cases} 0, & \text{if } |I'_k(i,j) - \mu_k| > t * \sigma_k \\ 1, & else \end{cases} \tag{6}$$

Where, μ_k represents the mean value and σ_k the standard deviation of I_k. This means O is a binary mask that covers the outliers from *both* dimensions. The usual values for tolerance factor t are in the range $[3, 10]$. Small values for t will lead to many pixels being regarded as outliers which means losing almost all sharp features. In the case of two objects which are partly occluding each other it may also occur that they appear to *melt*. A higher value for t, will cause a tolerance for larger steps which impairs the visibility of smaller features.

I''_k is obtained by setting the corresponding values to 0 like it is done above:

$$I''_k = O \odot I'_k \tag{7}$$

This eliminates unnecessary depth ranges and leads to continuous gradient images without jumps or discontinuities at the boundary and the scene objects. In this way, the outlier detection adapts automatically to scene elements without the need for absolute thresholding parameters.

In order to compress the initial height field we have to reduce the amplitude of the gradient signal. This is done by first applying an attenuation function which brings the entries closer together by diminishing larger values stronger and boosting small ones. In contrast to [5] who use a logarithmic weighting function which compresses the entries only by regarding their absolute value, we opted for applying the adaptive function proposed in [7] which takes into account the properties of the depth interval.

$$\begin{aligned} A(X,i,j) &= \begin{cases} 0, & if X(i,j) = 0 \\ \frac{a}{|X(i,j)|} \cdot \left(\frac{|X(i,j)|}{a}\right)^b, & else \end{cases} \\ I'''_k &= A(I''_k) \odot I''_k \end{aligned}$$

The parameter a is chosen to be 10% of the average absolute value of all unmasked foreground pixels, it tags values which map to 1 in the attenuation function. Pixels with entries whose absolute value is smaller than a are slightly enhanced whereas those above it are compressed. The second parameter b steers the attenuation rate. It is set to 0.9 for all results substantiated in this paper.

Now, that the gradient images are continuous and attenuated, the signals need to be decomposed and the relative importance of small details needs to be enhanced. Therefore, we use bilateral filtering, which is a well known technique in 2D image processing that performs edge-preserving smoothing. Using this filtering on the gradients is the core idea of our algorithm. Preserving gradient edges means preserving ridges.

The filter is described in the following equation:

$$\begin{aligned} BF(X,i,j) &= \frac{\sum_{m,n} W_{i,j}^{m,n}(X) X_{m,n}}{\sum_{m,n} W_{i,j}^{m,n}(X)} \\ W_{i,j}^{m,n}(X) &= G_{\sigma_s}(|| \begin{pmatrix} i \\ j \end{pmatrix} - \begin{pmatrix} m \\ n \end{pmatrix} ||) G_{\sigma_r}(|X_{i,j} - X_{m,n}|) \end{aligned}$$

Here, G_{σ_s} stands for a 1D Gaussian kernel with standard deviation σ_s which is used to describe how much influence the *spatial* distance has on the result, whereas G_{σ_r} steers how strong the difference of values (*range*) affects it. m and n range from 0 to the maximal resolution in X and Y respectively. For a detailed description we refer to [17] where several variations of this filter are investigated. The values for the deviation are chose adaptively, we recommend to use $\sigma_s = \frac{min(Res_X, Res_y)}{16}$ and $\sigma_r = \frac{X_{max} - X_{min}}{10}$, which are the default values for all results in this paper.

We now perform the decomposition in the following way:

$$Coarse_k = BF(I_k''') \tag{8}$$

$$Fine_k = I_k''' - Coarse_k \tag{9}$$

The new gradient components J_k for the final bas-relief are now generated by modifying the relation between the coarse and the fine components.

$$J_k = Fine_k + r * Coarse_k \tag{10}$$

This leads to penalizing the coarse level details and a relative boosting of the fine details and ensures that the smaller features will remain perceivable in the result. Typically the value for the relation $r \in [0.05, 0.25]$. $r = 0$ would lead to exaggerated results which only contain small features, spherical parts would appear either flat or noisy. $r = 1$ does not change the relation at all and it would be the same as linear recalling which is not intended as described earlier.

Given the new gradient $\nabla J = \begin{bmatrix} J_x \\ J_y \end{bmatrix}$, we now have to reconstruct its corresponding height field. In order to get back from the gradient domain to the spatial domain, we first compute its Laplacian ΔJ by adding the second derivatives in both dimensions. We are still in a discrete case, so this can be done using finite differences. Since ΔJ is defined by a central difference, we have chosen the backward difference for this case:

$$\Delta J = J_{xx} + J_{yy} \tag{11}$$

$$J_{xx}(i,j) \approx J_x(i,j) - J_x(i-1,j) \tag{12}$$

$$J_{yy}(i,j) \approx J_y(i,j) - J_y(i,j-1) \tag{13}$$

The computation of a function J given its Laplacian is a so called Poisson problem and it is a fairly standard technique which require solving a sparse system of linear equations.

3.4 Postprocessing

At the final stage, we proceed to the reassembly of the modified gradient components. As the boundary in the Poisson reconstruction is given by the "frame" of our normalized height field, J may contain positive as well as negative values. Therefore, a normalization is needed for setting the background and the unreliable values along the object boundary to 0 again:

$$\hat{J}(x,y) = B(i,j) \cdot S(i,j) \cdot (J(x,y) - J_{min}) \tag{14}$$

The depth entries of $\hat{J}$ now range form 0 to $\hat{J}_{max}$ (maximal foreground value of $\hat{J}$). In general this would not exactly match the desired interval size. In the last step a linear re-scaling to the correct ratio or range is performed.

$$\text{achieved range} = \hat{J}_{max} \tag{15}$$

$$\lambda = \frac{\text{desired range}}{\text{achieved range}} \tag{16}$$

$$\overline{J} = \lambda \cdot \hat{J} \tag{17}$$

In this way we have produced a flattened version $\overline{J}$ of the initial height field I which maps the smallest foreground value to the background plane and elevates the rest only in slight manner. Thanks to our detail enhancement technique, all the fine structure remain perceivable in the final result.

For visualization purposes, we use a triangle mesh, based on a regular grid, for which the number of vertices is equal to the depth map resolution, and displace every vertex by its corresponding height value.

4. Artistic Applications

The approach of cubism to painting consists of breaking up the traditional vision of reality into multiple perspectives which are combined in a single composition. The resulting images give the impression of being viewed from many different angles at once. Needless to say that cubism reflects a subtle aspect of human perception, which is reliance on more than a single glance.

We show how a slight modification of the above mentioned main procedure can help generating cubism like relief sculptures. We extend the height field capturing in a way that requires the user to rotate the model in frontal view, then a sequence of 13 height fields form -90 degree to 90 degree, which differ in 15 degree each, is automatically captured.

The user can now cut and paste multiple desired perspectives into a new height field with any standard image editing tool. The problem which arises is that the length of the depth interval may vary throughout the different perspectives and also might be affected by the visibility of certain parts. These changes are rather large in general, as illustrated by the color coding in Figure 2. Nevertheless, to our advantage these differences cause large discontinuities along the transition areas of two or more perspectives, such that our gradient outlier detection delivers those areas for free.

As described above, we set the corresponding large gradient values to 0. The remaining issue is that after reconstruction, the bas-relief exhibits a visible seam because a null gradient leads to a flat transition with steps on both sides which even emphasize the impression of two distinct parts. Our experiments revealed that using a diffusion process or blurring those pixels in the spatial domain (after reconstruction) may lead to even worse results, as they introduce additional steps between modified and unmodified entries.

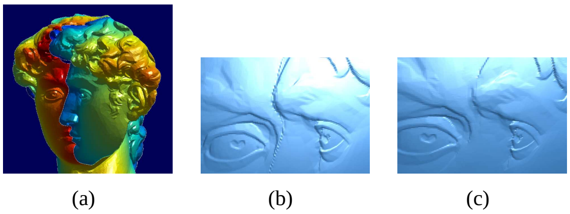


Fig. 2. (a) Color coded depth of the assembled shape; (b) zoom on the back of the nose of relief without seam treatment; (c) improved result; full relief can be seen in Figure 1

Since the reconstruction requires central difference before reconstruction, each pixel which has an outlier in its direct neighborhood will be affected. To overcome this problem we detect all affected gradient locations and use a Gaussian filter in the gradient domain in order to get smooth transitions which finally lead to a geometrically seamless result. We extend the outlier mask by adding all pixels which are situated next to an outlier. Therefore we convolve O with a 3x3 kernel:

$$F = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \tag{18}$$

$$\hat{M} = O \otimes F \tag{19}$$

$$M(i,j) = \begin{cases} 1, & \hat{M}(i,j) = 9 \\ 0, & else \end{cases} \tag{20}$$

This mask M marks the positions whose value after reconstruction is still not reliable because of the outlier removal. As we need to exclude the other 0-entries for the blurring, we rely on the following discrete convolution and modify the new gradient components J_k accordingly:

$$\begin{aligned} D_\sigma(X,i,j) &= \frac{\sum\limits_{m,n} G_{\sigma_b}\text{(m,n)M(i-m,j-n)X(i-m,j-n)}}{\sum\limits_{m,n} G_{\sigma_b}\text{(m,n)M(i-m,j-n)}} \\ J'_k(a,b) &= D_\sigma(J_k,a,b) \\ \forall a,b &: M(a,b) = 0 \end{aligned}$$

In this case, G_{σ_b} is a 2D Gaussian kernel and m, n are its indices. For all cubism results presented in this paper σ_b is set to 8. As a straightforward extension, a user can even drag and drop height fields of very different models with largely varying spatial extensions into one large *geometric collage*, and the modified algorithm will produce a bas-relief sculpture without nasty transition areas. The tool itself is very tolerant, flexible and offers a lot of freedom to the user, but a meaningful creation of the input is mandatory for generating visually pleasing results, this depends on the skill of the user in arranging the different perspectives or objects.

5. Results and Discussion

All results presented here contain a slightly elevated silhouette mainly due to the final normalization step. This outline exaggeration is also used by real sculpting artists, to give the artwork a life-like impression and it does not seem to be sunken in the background. If this is not desired, or harms the depth range too much, a Gaussian smoothing along the boundary can be used as further postprocessing. This is straightforward since the outline location is already known.

All models were compressed so that the depth range is equal to 1% of their largest dimension (X or Y). The most crucial part for generating high quality bas-reliefs is the resolution of the discrete height field. On the one hand, a low resolution yields not so fine details and on the other hand it carries the risk that the outlier recognition is not representative anymore as too many pixels may get mistakenly disregarded. Moreover, if a surface has details along its silhouette which possess a width of only 1 pixel, then they may get removed by the silhouette detection. To overcome these issues, the resolution need to be sufficiently high.

We want to stress that the outlier removal is not always necessary, but depending on the model and purpose, it turns out to be helpful. However, for the generation of seamless cubism like reliefs it is absolutely mandatory, because it removes the steps along the transition areas.

We were kindly provided with two relief models by the authors of [5] which we use for comparison. We did not have access to an implementation but we tried our best to acquire the same model pose and conditions to ensure a fair comparison. The outcomes of [4] were achieved by their implementation with exactly the same input as it was used for the presented algorithm. Figure3 compares our results to those of other to dates methods.

In these cases the approach of [4] seems to exaggerate the small features. Spherical parts like the eyeball of the dragon appear to be flat and not reproduced well. The models from [5] exhibit some problems with bumps as illustrated by the highlights. This is due to the interplay between thresholding, manipulation and the reconstruction step. The authors propose a method which helps to overcome this problem on the objects silhouette. Nevertheless, this will not handle the artifacts on the castle entrance or the inner roofs. Also in that approach the features in the foreground of the dragon relief are visible much better than those in the background. The quality of our features is constant everywhere and the dragons claws appear more natural in our case.

Figure 4 shows one more models compressed with our method as well as a cubism result with blurring of the affected entries and another montage consisting of 4 different objects: a greek statue, a mask of a pharaoh, a bunny and a cup.

In the case of a collage, a similar result could be achieved by arranging the objects in a 3D scene, but if the height fields are already given, an artist can place them with a regular image editing tool much easier, and our algorithm would automatically remove undesired discontinuities.

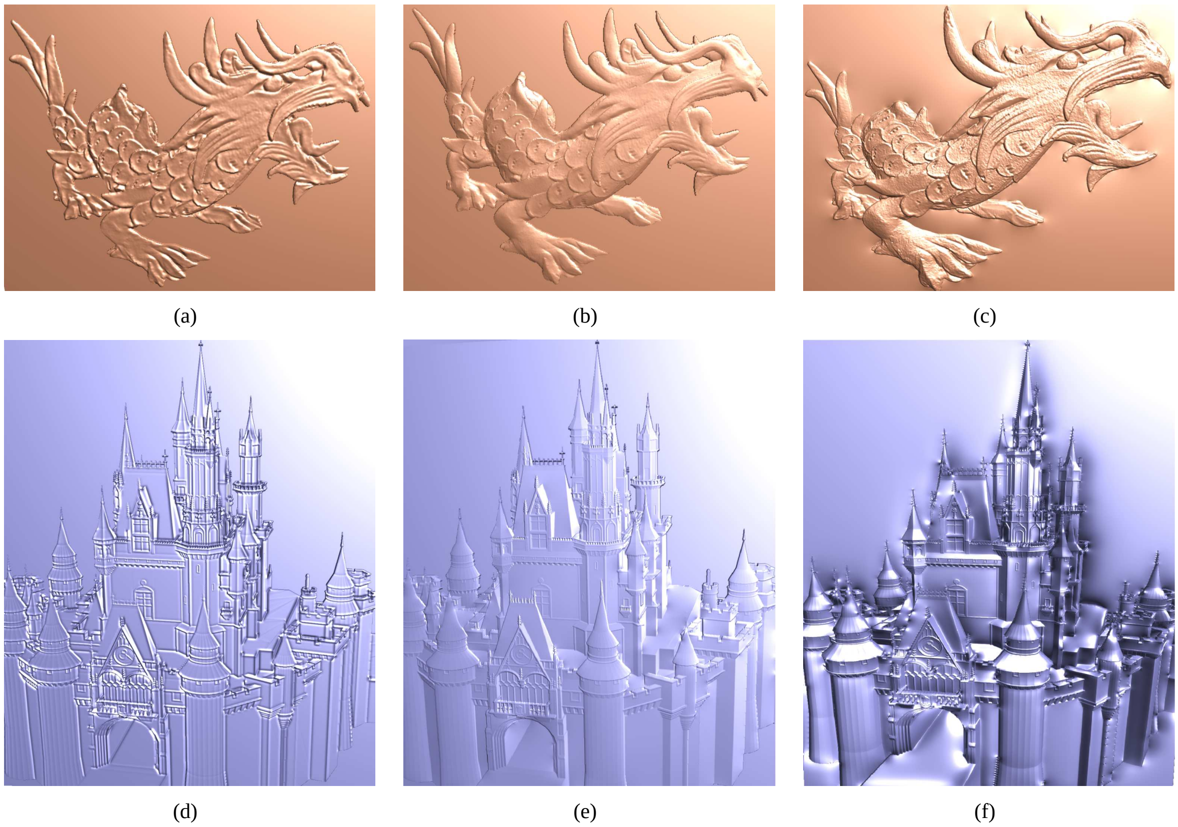


Fig. 3. (left) results of the appraoch of Kerber et al., (middle) reliefs achieved with the presented approach, (right) results of the approach of Weyrich et al.

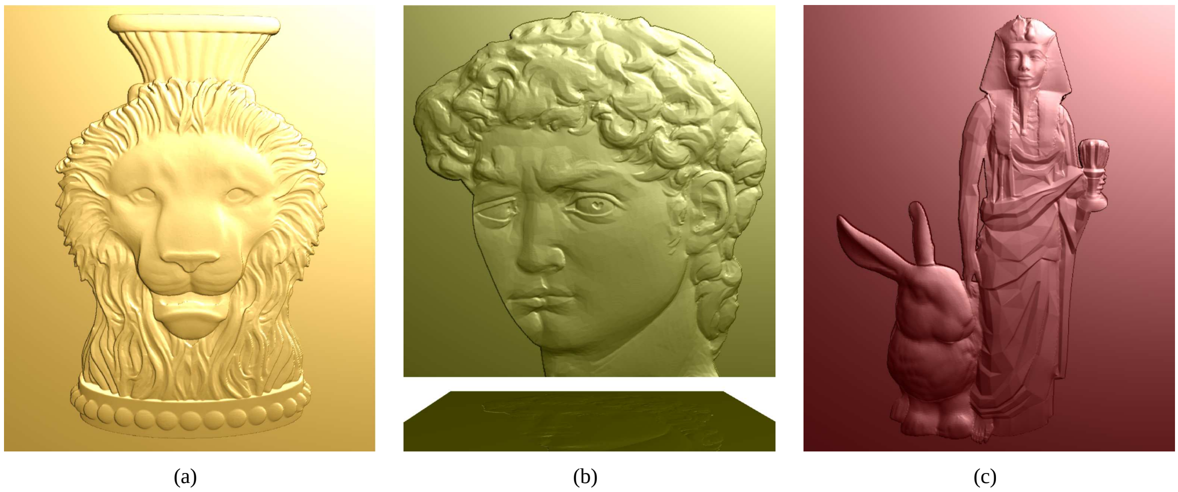
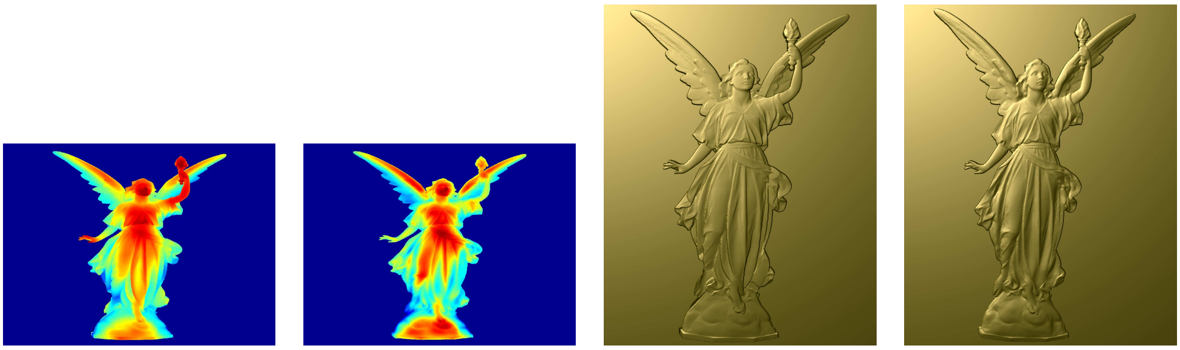


Fig. 4. (a) Relief of the lion vase model(b) another cubism like effect on the David model (c) a collage being assembled of 4 different objects

Figure 5 demonstrates the difference between applying the bilateral filter in the spatial domain and using it to filter in the gradient domain, as it is done here. Preserving edges like it is done in the spatial case is strictly speaking counter productive for our compression purpoes because these edges are not visible form na orthogonal point of view and they cover keeping them keeps the depth interval size unnecessarily high. (the interval size for the Lucy model dropped from 407 to 90 for the spatial case compared to 52 in the gradeint case, before linear sclaing was applied). You can see this difference in the transition between the raised arm and the wing. The overall impression is quite ok but not as pleasing as the one achieved with our approach. Note the problems at the left foot in the spatial result and compare the richness of fine details of on the wings, the torch and also the fingers in the two results.

A shortcoming of our approach is that it does only one decomposition. In [5] a multi level approach is used in order to allow stop band filtering. Such ideas are helpful because one may allow a better distinction between very fine details and noise.



(a) (b) (c) (d)

Fig. 5. Colorcoded height fields and the rendered reliefs achieved when filtering in the spatial (a+c) and the gradient domain (b+d)

Besides from the intended compression ratio, our method requires only two input parameters from the user (at most). This makes our approach more attractive in comparison to existing approaches which may require a *trial and error* tactic for setting the proper weights for the multiple layers of the Gaussian pyramid [5] in addition to a threshold which can vary from model to model.

6. Conclusion

We presented a semi-automatic tool intended to support the creation of bas relief from virtual scenes. The key technical contribution of our work is a filtering approach which is aimed at preserving curvature extrema during the compression process. In this way it is possible to handle complex scenes with fine geometric detail. Furthermore, we have simplified the relief generation process so that it hardly requires any user intervention. On the artistic side we demonstrated how to use our technique for generating cubism based bas relief scenes. The whole framework is intuitive, easy to implement and and independent of scene complexity.

Acknowledgements

We would like to thank the Stanford 3D Scanning Repository, Google 3D Warehouse, the XYZ RGB Inc. and AIM@SHAPE for providing the models used throughout this paper. Special thanks to Tim Weyrich and Jia Deng for granting access to their results.

References

[1] Peter N. Belhumeur, David J. Kriegman, and Alan L. Yuille, “The bas-relief ambiguity”, *Int. J. Comput. Vision*, vol. 35, no. 1, pp. 33–44, 1999.

[2] Paolo Cignoni, Claudio Montani, and Roberto Scopigno, “Automatic generation of bas- and high-reliefs”, *Journal of Graphics Tools*, vol. 2, no. 3, pp. 15–28, 1997.

[3] Wenhao Song, Alexander Belyaev, and Hans-Peter Seidel, “Automatic generation of bas-reliefs from 3d shapes”, in *SMI ’07: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2007*, Washington, DC, USA, 2007, pp. 211–214, IEEE Computer Society.

[4] Jens Kerber, Alexander Belyaev, and Hans-Peter Seidel, “Feature preserving depth compression of range images”, in *Proceedings of the 23rd Spring Conference on Computer Graphics*, Mateu Sbert, Ed., Budmerice, Slovakia, April 2007, pp. 110–114, Comenius University, Bratislava.

[5] Tim Weyrich, Jia Deng, Connelly Barnes, Szymon Rusinkiewicz, and Adam Finkelstein, “Digital bas-relief from 3d scenes”, *ACM Trans. Graph.*, vol. 26, no. 3, pp. 32, 2007.

[6] Paul Debevec and Erik Reinhard, “High-dynamic-range imaging: Theory and applications”, SIGGRAPH 2006 Course #5, 2006, http://www.siggraph.org/s2006/main.php?f=conference&p=courses&s=5.

[7] Raanan Fattal, Dani Lischinski, and Michael Werman, “Gradient domain high dynamic range compression”, in *SIGGRAPH ’02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 2002, pp. 249–256, ACM.

[8] Jens Kerber, “Digital art of bas-relief sculpting”, Masters thesis, Universität des Saarlandes, August 2007.

[9] Xianfang Sun, Paul L. Rosin, Ralph R. Martin, and Frank C. Langbein, “Bas-relief generation using adaptive histogram equalisation”, *IEEE Transactions on Visualization and Computer Graphics*, 2009.

[10] Frédo Durand and Julie Dorsey, “Fast bilateral filtering for the display of high-dynamic-range images”, in *SIGGRAPH ’02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 2002, pp. 257–266, ACM.

[11] Andrew S. Glassner, “Cubism and cameras: Free-form optics for computer graphics”, Tech. Rep. MSR-TR-2000-05, Microsoft Research, January 2000.

[12] Jingyi Yu and Leonard McMillan, “A framework for multiperspective rendering”, in *Rendering Techniques*, Alexander Keller and Henrik Wann Jensen, Eds. 2004, pp. 61–68, Eurographics Association.

[13] Andrew Adams and Marc Levoy, “General Linear Cameras with Finite Aperture”, Grenoble, France, 2007, pp. 121–126, Eurographics Association.

[14] Alexander A. Pasko, Vladimir Savchenko, and Alexei Sourin, “Synthetic carving using implicit surface primitives”, *Computer-Aided Design, Elsevier*, vol. 33, no. 5, pp. 379–388, 2001.

[15] Alexei Sourin, “Functionally based virtual computer art”, in *SI3D ’01: Proceedings of the 2001 symposium on Interactive 3D graphics*, 2001, pp. 77–84.

[16] Fábio Policarpo, Manuel M. Oliveira, and ao L. D. Comba Jo “Real-time relief mapping on arbitrary polygonal surfaces”, in *SIGGRAPH ’05: ACM SIGGRAPH 2005 Papers*, New York, NY, USA, 2005, pp. 935–935, ACM.

[17] Sylvain Paris and Frdo Durand, “A fast approximation of the bilateral filter using a signal processing approach”, Tech. Rep. MIT-CSAIL-TR-2006-073, MIT, 2006.