Special Issue: CAD in the Arts

Guest Editor: Carlo H. Séquin, University of California, Berkeley

Real-time Generation of Digital Bas-Reliefs

Jens Kerber[1], Art Tevs[1], Alexander Belyaev[2], Rhaleb Zayer[3] and Hans-Peter Seidel[1]

[1]MPI Informatik, Saarbrücken, Germany
[2]Heriot-Watt University, Edinburgh, UK
[3]INRIA, Nancy, France

ABSTRACT

Bas-relief is a form of sculpture where carved or chiseled forms protrude partially and shallowly from the background. Occupying an intermediate place between painting and full 3D sculpture, bas-relief sculpture exploits properties of human visual perception in order to maintain perceptually salient 3D information. In this paper, we present two methods for automatic bas-relief generation from 3D digital shapes. Both methods are inspired by techniques developed for high dynamic range image compression and have the bilateral filter as the main ingredient. We demonstrate that the methods are capable of preserving fine shape features and achieving good compression without compromising the quality of surface details.
For artists, bas-relief generation starts from managing the viewer's point of view and compositing the scene. Therefore we strive in our work to streamline this process by focusing on easy and intuitive user interaction which is paramount to artistic applications. Our algorithms allow for real time computation thanks to our implementation on graphics hardware. Besides interactive production of stills, this work offers the possibility for generating bas-relief animations. Last but not least, we explore the generation of artistic reliefs that mimic cubism in painting.

Keywords: bas-relief, range compression, feature enhancement, bilateral filtering.
DOI: 10.3722/cadaps.2010.465-478

1 INTRODUCTION

1.1 Problem Setting, Challenges, and our Contribution

Similarly to paintings, the underlying medium for bas-reliefs is inherently two dimensional. The artistic importance of a bas-relief sculpture lies in conveying a volumetric projection of the design shapes into the viewer's space as if they detach themselves from the two dimensional background, as illustrated in Fig. 1 left. Bas-relief artists manage to suggest this 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 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.

Research on digital bas-reliefs is concerned with the automatic generation of artistically relevant bas-reliefs from 3D input scenes. The key ingredient in this process consists of compressing the so-called height field (also known as depth map or range image) which is a 2.5D description of the scene that encodes distance information on a 2D regular grid. Unfortunately, straightforward re-scaling of the height field fades perceptually salient shape features. Therefore a sophisticated and intelligent depth compression procedure is required to achieve plausible and impressive results.

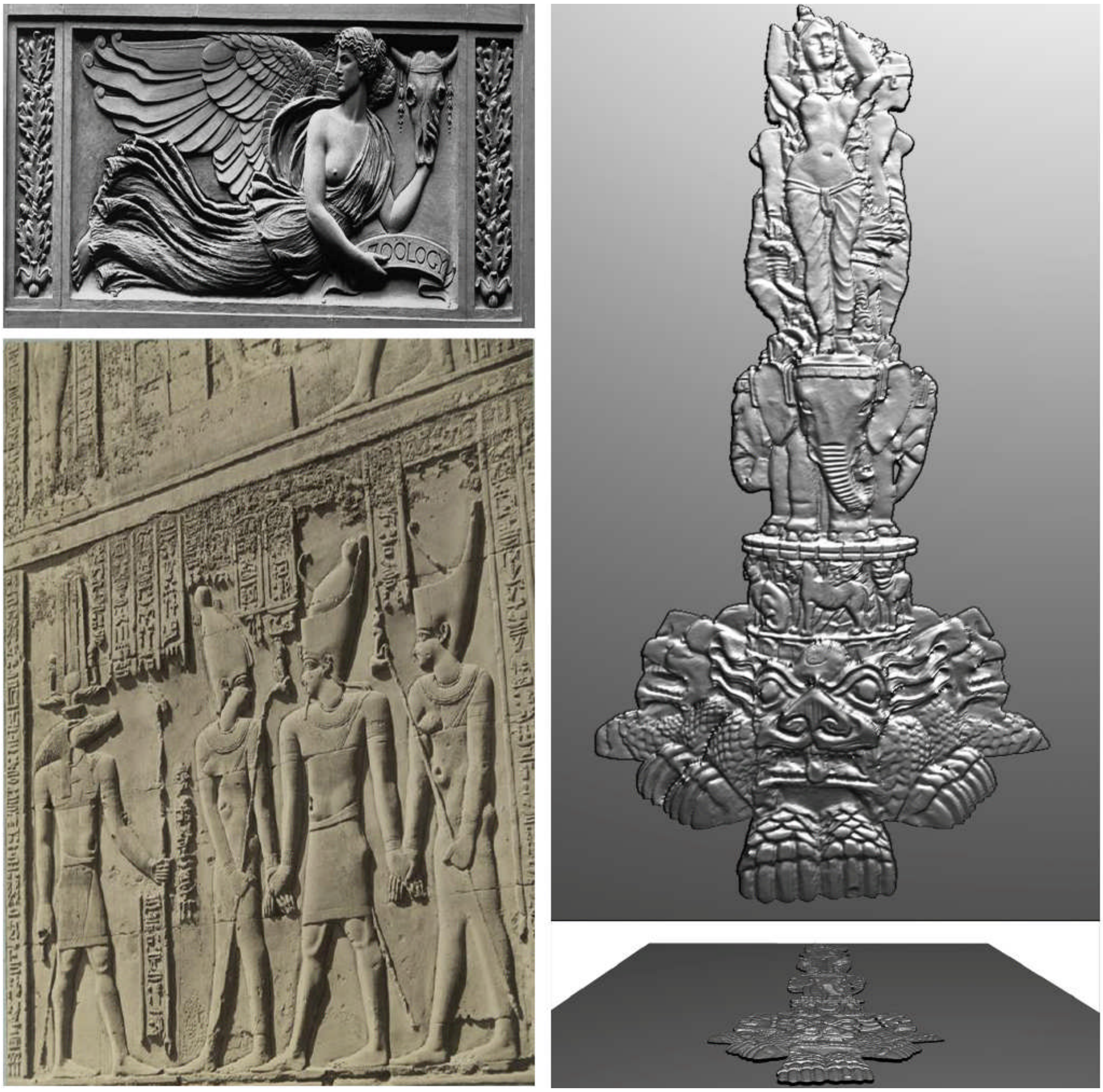

Fig. 1: (left) Two different examples of real-world artistic bas-reliefs. (right) A digital bas-relief of the Thai-statue generated with our gradient domain method and a tilted view on it, to illustrate its flatness.

Many of the existing bas-relief generation techniques [4], [6], [7], [11], [12], [14] differ mainly in the compression step. Many of these methods can achieve results of impressive quality but often the number of user-specified parameters involved and computational complexity makes them labor intensive. 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 significant compression without compromising the quality of surface details. We present two solutions to this problem, the first operates in the range domain and the second works in the gradient domain. Both methods take advantage of bilateral filtering. Our approaches are conceptually fairly simple and can be easily implemented on graphics hardware.

Certainly, the compression step is a fundamental step to the bas-relief generation process as it should preserve the richness of details and the visibility of important features in the flattened result. Nevertheless, it is worthwhile to note that from an artistic vantage point, the design process of bas-reliefs starts much earlier at the level of scene compositing and managing the viewer's point of view which are equally important for achieving the desired design goals. So far, this problem has been studied less, and digital artists have to go through lengthy iteration cycles involving all these steps before they can get satisfactory results. In order to streamline this pipeline, we developed an artistic tool that features real time semi-automatic bas-relief generation with only a small number of intuitive yet artistically relevant control parameters. Besides the interactive design of stills, our method is the first one capable of generating bas-relief animations in real time. In the current setup, animation related aspects such as frame coherence are directly inherited from the 3D animation. Our experiments suggest that this solution is sufficient in practice.

To further explore the artistic possibilities of digital compositing, we investigate the impact of multiple perspectives on the bas-relief generation. For this purpose, we study the problem of merging different perspectives into one single scene, an intriguing style pioneered by cubist painters such as Picasso.

1.2 Related Work

Although a bas-Relief is a very flat sculptural representation, it suggests showing a fully extended 3D scene. The depth effect is achieved by a combination of shading and shadowing which is similar to the original conditions. This phenomenon of human perception was investigated in [3]. If the vantage point differs too much from an orthogonal view, then the illusion will be revealed. The existence of this so called bas-relief ambiguity becomes a challenge in other disciplines like machine vision. In this research area, the task is to estimate a model’s shape by recording it from several viewpoints using a regular camera. Due to the bas-relief ambiguity, this estimation is not unique in general, unless a user makes prior assumptions about the shading properties of the object’s surface [2], [13].

The first achievements in automatic relief generation were made by [4]. The authors use perspective foreshortening which keeps closer objects more salient than objects situated further away. Their compression step consists of simple linear re-scaling. Although this work has addressed the main aspects of relief generation, the visibility of fine structural details suffers for the benefit of rough surface features.

More recent works in this field developed techniques that achieve depth compression without losing visually important information. Most of the recent work can be regarded as variants of tone mapping approaches [5] which aim to compress the very large luminance interval of high dynamic range images so that they can be displayed on regular monitors without losing contrast impression. However, modifying these techniques to fit the needs of a geometric compression task is far from straightforward, because one has to deal with problems like foreground-background-transition and self-occlusions.

The algorithm proposed in [11] works with a saliency measure and an appropriate enhancement procedure. They rely on differential coordinates, and the final relief is achieved by a diffusion process. The methods described in [6] and [14] operate in the gradient domain. In [6] the authors extract high frequency signals from the gradient by applying a Gaussian filter. This technique is not entirely capable of preserving the sharpness of surface features and the results appear to be exaggerated. In contrast to this, a silhouette respecting diffusion filter is applied in [14]. More artistic freedom is added by allowing a user to individually boost a number of different frequencies such that their presence in the outcome is enhanced or decreased. The results are impressive but their method requires tuning a lot of parameters. In the latter two cases a sparse system of linear equations has to be solved in order to reassemble the result from its manipulated gradient field.

The method recently presented in [12] adapts a histogram based contrast enhancement technique applied directly to the height field. Gradient information is only used additionally. The results are very pleasing but unfortunately the method is time consuming.

Bas-reliefs can be used as templates for real world applications like engraving, embossing and milling, or serve as input for further applications in computer art [9] and virtual shape decoration [10].

1.3 Paper Organization

The rest of this paper is organized as follows. In Sections 2 and 3, we describe in detail the algorithms of our range and gradient domain approaches to bas-relief generation, respectively. The range domain technique is an extension of our earlier work [7] which is summarized by the description of the gradient method. Artistic applications to merging multiple perspectives are presented in Section 4 and a description of our interactive bas relief application and its performance are given in Section 5. Finally the various generated bas-reliefs are compared to each other in Section 6.

2 RANGE DOMAIN APPROACH TO BAS-RELIEF GENERATION

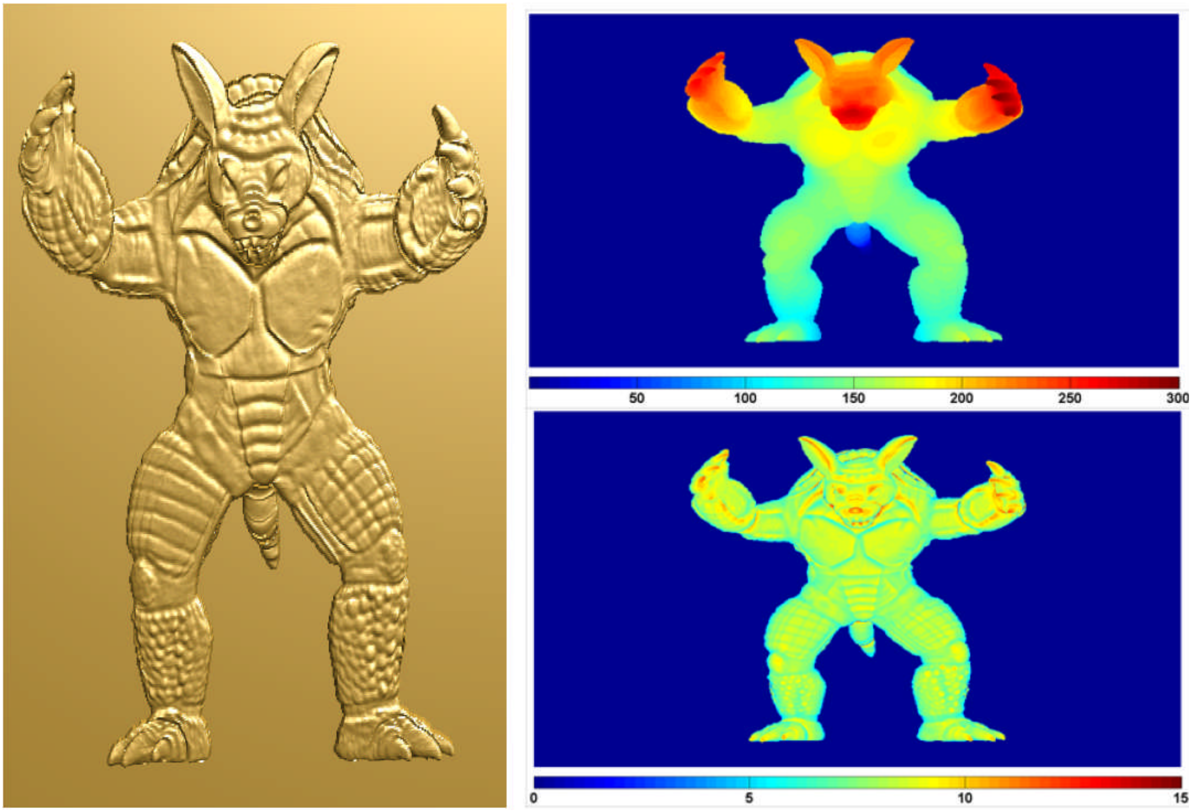

Fig. 2: Bas-relief of the armadillo model obtained with our range domain approach (left) along with the original depth map (right top) and the bas-relief height field (right bottom); note the different interval ranges.

2.1 Preprocessing

Starting from the 2D depth map $\bar{I}(i) = d$ of the input scene, we extract a binary background mask B by checking it against a default background value δ. This mask is further used to normalize the scene by bringing its height values into the interval $[0, R_0]$ in the following manner.

$$B(i) = \begin{cases} 0, & if\ \bar{I}(i) = \delta \\ 1, & otherwise \end{cases}$$

$$I(i) = B(i) \cdot (\bar{I}(i) - \min(\bar{I}))$$

$$R_0 = \max(I)$$

2.2 Decomposition and Enhancement

Bilateral Filtering - In the current depth map, self-occlusions and large discontinuities on the surface are reflected as edges. Although, these edges are not visible from an orthogonal vantage point they cover a huge amount of the entire initial depth range, and therefore offer a high compression potential. As these edges contain important information about the depth order of the different patches, preserving this information during the compression process is essential for inferring a more plastic impression to the resulting bas-relief.

In our approach, we compress the spatial extent of the height field by extracting a partly linear base image which encodes the coarse structure of the underlying shape. For this purpose, we use a bilateral filter that permits edge-preserving smoothing to extract the base layer of the shape. The filtering process is done by convolution with the product of two Gaussian kernels. One of them weighs the spatial distance of pixels in a local neighborhood, whereas the other one penalizes deviations in the depth values of the entries. By extending the standard bilateral filter by a binary mask, we can guarantee that the value for background pixels is zero by default and that they do not influence the computation of foreground patches.

$$BF(X,M)(i) = \frac{M(i) \cdot \int\limits_{j \in X} G_{\sigma_S}(i-j) \cdot G_{\sigma_R}(X(i) - X(j)) \cdot M(j) \cdot X(j)\, dj}{\int\limits_{j \in X} G_{\sigma_S}(i-j) \cdot G_{\sigma_R}(X(i) - X(j)) \cdot M(j)\, dj}$$

$$I_{\text{base}} = BF(I, B)$$

$$I_{\text{detail}} = I - I_{\text{base}}$$

Here, G_{σ_k} stands for a Gaussian kernel with appropriate standard deviation. This equation contains two user defined parameters, namely σ_S and σ_R. With the help of σ_S, an artist can control how much the surroundings of a pixel can influence its entry. Small values for σ_R lead to a large number of *iso-levels* in I_{base}, whereas a higher tolerance leads to coarser details which may remain in I_{detail}. In practice, σ_R is chosen as a fractional amount of R_0. For further details about bilateral filtering, we refer to [8].

Laplacian-like diffusion - In a bas-relief representation, the viewer's visual experience is enhanced through the subtle elevation and depth implied by the ridges and valleys of the shape. Those are the areas in which the curvature reaches local extrema. In order to better capture these important features, we note that the detail layer, which we gained from the prior step, contains information about the constitution of the object’s surface. This information itself consists of low and high frequency components. Thus we further split the detail layer into a coarse and a fine layer.

For this purpose, we opted for a Laplacian-like diffusion. The Laplacian operator is widely used for image denoising because entries with a high curvature are smoothed more strongly. Since background pixels must not have any influence on the computation of the Laplacian at the foreground, we have to pay special attention for the pixels at the boundary of the scene. In addition we have to suppress the influence of the edges separating the foreground and background and those corresponding to the foreground self-occlusions. One simple remedy is to use a modified Laplacian $\tilde{\Delta} = \Delta / (1 + |\nabla|^2)$ where ∇ stands for the gradient. One can see that the factor $1/(1+|\nabla|^2)$ restricts the influence of these edges. In a discrete representation, the smoothing process reads as follows:

$$\tilde{\Delta}(X, M)(i) = M(i) \cdot \frac{\sum_{n \in N(i)} M(n)X(n) - X(i) \cdot \sum_{n \in N(i)} M(n)}{1 + |\nabla X(i)|^2}$$

$$I_{\text{coarse}} = \begin{cases} I_{T=0} = I_{\text{detail}} \\ I_{T=T+1} = I_T + \alpha \cdot \overset{\%}{\Delta}(I_T, B) \end{cases}$$

$$I_{\text{fine}} = I_{\text{detail}} - I_{\text{coarse}}$$

Here $N(i)$ stands for the four direct neighbors of pixel i in the X and Y directions. In case that all neighbors belong to the foreground, the numerator in the equation is identical to the standard Laplacian expression. Here, we are given two more parameters that influence the computation. The first parameter is the time step size α which controls how fast the diffusion reaches a steady state. It should not be larger than one quarter of the pixel spacing because otherwise the approximation would become unstable. Smaller values lead to more accurate but slower approximation. Another user control is given by the number of diffusion iterations. This affects how strong the smoothing will actually be.

Unsharp masking - Now that the fine layer contains the high-frequency small-scale features of the surface, we will further boost their influence relative to the low frequency details in the coarse layer. At the same time we rescale the base layer linearly such that the importance of the detail layer is further enhanced. This results in a new height field with adjusted weights for the different layers. Higher values for both enhancement parameters imply that more details will remain perceivable in the relief R. For $\lambda_1 = \lambda_2 = 1$ the result would be the same as a linear rescaling of the input height field.

$$R = \frac{1}{\lambda_1} I_{\text{base}} + I_{\text{coarse}} + \lambda_2 I_{\text{fine}}$$

2.3 Post-processing

The intermediate result is now normalized like it was done with the input depth map, because the boundaries of the range interval might have changed drastically. We also measure the spatial extent that we achieved so far. Then, we linearly rescale the result either to the desired compression ratio or to the allowed depth range.

$$R(i) = B(i) \cdot (R(i) - \min(R))$$

$$R_1 = \max(R)$$

$$\mu = \frac{R_{\text{desired}}}{R_1}$$

$$Basrelief = \mu \cdot R$$

The spatial extent of this final result has now dropped to the user defined height. Thanks to the enhancement of the smaller features, they remain perceivable in the Bas-Relief.

3 GRADIENT DOMAIN APPROACH TO BAS-RELIEF GENERATION

In the previous section, we capitalize on the fact that surface discontinuities can be regarded as edges in the depth map. In the current approach, the guiding idea is that in the gradient domain, surface ridges are the edges of the gradient field. Since the initial and final normalization of height values are independent of the method, the pre- and post-processing steps required by our gradient domain approach are identical to the ones described in the previous section. Therefore we will omit them here and directly delve into the details of our compression approach.

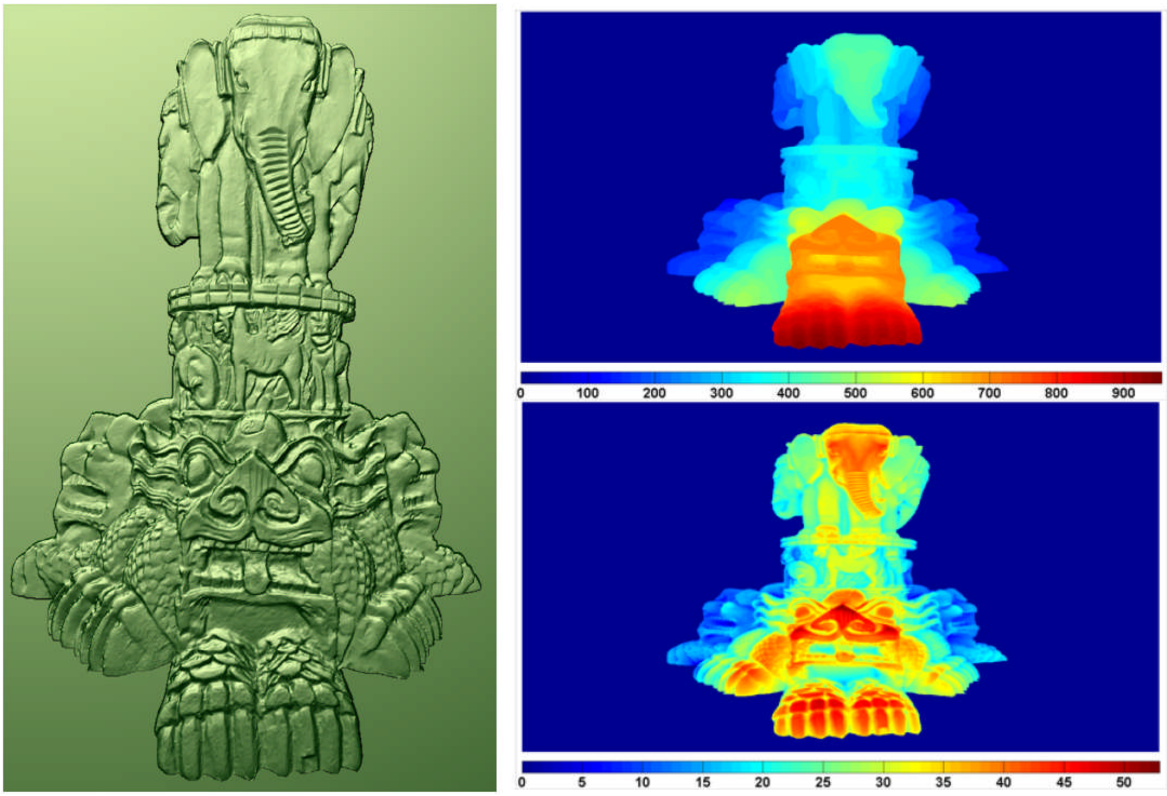

Fig. 3: Bas-relief of the XYZRGB Thai-statue obtained through the gradient domain approach (left) along with the original depth map (right top) and the bas-relief height field (right bottom); note the different interval ranges.

Starting from the normalized height field I and the background mask B (obtained after pre-processing). We first compute the gradient images I_x and I_y of I.

$$I_x = \frac{\partial I}{\partial x}$$

$$I_y = \frac{\partial I}{\partial y}$$

In general, this gradient field exhibits high jumps at boundary regions and its occlusions areas, therefore a special treatment is need to avoid this singular behavior.

Detecting issues - As the background values are generally very different from the foreground entries, the outlines of the scene objects will have rather large gradients. We can detect these silhouettes by computing a binary image S with the help of the background mask gradients which have lengths equal to one when a pixel belongs to an affected area. The gradient information at those positions is set to zero. In order to keep this exposition concise we use the notation $k \in \{x,y\}$ from now on.

$$B_k = \frac{\partial B}{\partial k}$$

$$S_i = \begin{cases} 0, & if\, B_k(i) = 1 \\ 1, & otherwise \end{cases}$$

$$I_k = I_k \,\mathrm{o} S$$

Here the $\circ$ operator stands for component-wise multiplication.

In general, the resulting shape also exhibits self-occlusions and large jumps on its surface which would negatively affect the quality of the result. On the one hand, they would keep the depth interval size artificially high; on the other hand large features would drastically impair the visibility of smaller details. Therefore, we rely on an outlier detection to locate gradient entries that differ largely from the other ones. Those positions are stored in an outlier mask O which is employed to set corresponding entries to zero similarly to the previous step.

$$O(i) = \begin{cases} 0, & if\, |I_k(i) - \mu_k| > t \cdot \sigma_k \\ 1, & otherwise \end{cases}$$

$$I_k = I_k \circ O$$

Here, μ_k represents the mean value, and σ_k stands for the standard deviation of all foreground pixels in I_k . Note that O covers outliers from *both* dimensions. The tolerance parameter t allows controlling the strictness of this criterion.

Unsharp masking - Thus far, we have generated continuous gradient images without jumps or discontinuities at the boundary and on the surface. In order to enhance the relative importance of high frequency details we need to decompose the surface signals. As we are operating in the gradient domain, bilateral filtering is a candidate of choice. Its edge preserving properties translates to ridge preservation as ridges are naturally the edges of the gradient field. In this way, the filtering preserves the sharpness of details. In contrast, a regular blurring would wash out details. During bilateral filtering we disregard all pixels that have been detected previously as singularities.

$$M = B \circ S \circ O$$

$$I_{\mathrm{coarse}_k} = BF(I_k, M)$$

$$I_{\mathrm{fine}_k} = I_k - I_{\mathrm{coarse}_k}$$

We now compute the new gradient components R_k of the final relief by boosting the high-frequency small-scale details relative to the coarse ones, where the amount of enhancement is controlled by a user defined parameter λ .

$$R_k = I_{\mathrm{fine}_k} + \frac{1}{\lambda} I_{\mathrm{coarse}_k}$$

Reconstruction - We now have to reconstruct a height field, given the new gradients. In order to return from the gradient domain to the spatial domain, we first compute the Laplacian ΔR by adding the second derivatives in both dimensions.

$$\Delta R = \frac{\partial R_x}{\partial x} + \frac{\partial R_y}{\partial y}$$

The computation of a function, given its Laplacian, is known as a Poisson problem, and it amounts to solving a sparse system of linear equations, see e.g. [6, 14]. At this stage we proceed as in the post-processing in the previous section.

4 ARTISTIC APPLICATIONS

The simple setup of a single object or several non-overlapping objects is generally satisfactory for standard bas-relief applications. In some special situations, (e.g. conveying movement, storytelling, complex scenes...) much more artistic freedom is required. In such cases, the scenes can be composed in a 3D modeling program or simply by assembling a number of depth maps with the help of an image editing tool. In order to study this problem we explore the use of multiple perspectives within a single scene. In painting, this is known as cubism. Our goal then is to generate cubism-like bas-reliefs.

In this setting, the viewer should get the impression of watching this object from different viewpoints simultaneously. The technical challenge here lies in the fact that height fields stemming from different views may exhibit large differences in their depth interval range. These differences would be reflected in large jumps along the areas where two or more elements overlap. The windfall of our gradient domain approach is that the outlier detection finds these transitions automatically throughout the pipeline without the need for further processing steps. The only remaining challenge is that setting those areas to zero results in a flat transition which emphasizes the impression of having distinct scene elements, and this would definitely impair the desired impression. A simple remedy consists of slightly modifying the gradient domain approach to produce seamless results which maintain the cubism illusion. For this purpose, we proceed by filling all pixels that have at least one detected outlier in their direct surrounding with meaningful values before the reconstruction step. We construct an additional transition mask T which marks affected entries. After that, we use Gaussian filtering with a large support to fill these pixels. For this smoothing we ignore other masked pixels, of course.

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

$$T(i) = \begin{cases} 1, & if\, O \otimes F(i) = 9 \\ 0, & otherwise \end{cases}$$

$$R_k(i) = \frac{\int_{j \in R_k} G_{\sigma_S}(i-j) \cdot M(j) \cdot R_k(j)\,dj}{\int_{j \in R_k} G_{\sigma_S}(i-j) \cdot M(j)\,dj} \quad \text{for all } i \text{ s.t. } T(i) = 0$$

In Fig. 4 we illustrate the problem and the effect that this step has on the result. This example of the David head was achieved by first capturing two depth maps from different perspectives (90 degrees apart) and later arranging them in an image editing tool by using two different layers and exploiting transparency. As the color coding shows, the depth interval length is very different where the perspective changes (Fig. 4a). If the outliers are not removed a seam remains visible (Fig. 4b), unless a smoothing is applied to the problematic regions (Fig. 4c). One can also combine very different models with largely varying spatial extensions into one *geometric collage*, and the modified algorithm will automatically produce a bas-relief sculpture without nasty transition areas. Nevertheless, 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. As an example, we arranged three different models, namely the lion-vase, the Caesar head and the bunny model, with the help of a 3D modeling tool (Fig. 4f). We want to stress that the head and the vase do not touch in 3D. The Caesar model is simply placed right in front of it without editing any of the meshes. Finally the bunny is simply translated such that it appears to be situated on the top of the vase in this fixed perspective. The result shows that the described method nicely blends together the outlines of different models with different depth ranges and thus supports the impression of watching a continuous scene.

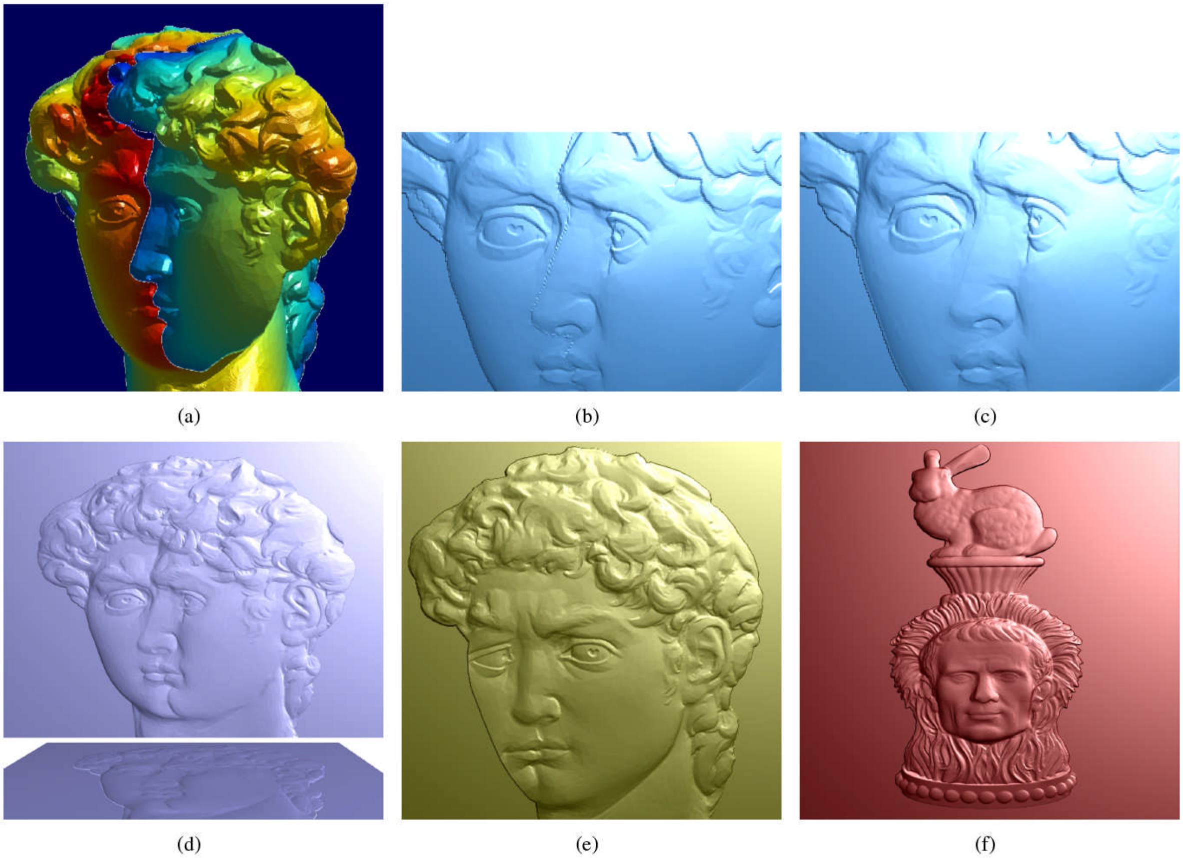

Fig. 4: (a) A color coded height field of the input; zoom on the result without (b) and after using (c) additional smoothing; (d) full size result and a tilted view which demonstrates its flatness; (e) another multi perspective effect on the David head; (f) a collage consisting of three different objects without any disturbing transitions.

5 AN ARTISTIC TOOL

5.1 Implementation

As a general remark, in most of previous work the depth map capture and the processing pipeline are not connected. This led to static applications, which require tedious work from the user to generate aesthetically pleasing results. Every time the user decides to use a different pose of the model, he is required to capture and store new height fields as well as tweak the appropriate parameter values by trial and error.

In our framework, we capitalize on the highly parallel nature of our method and exploit the properties of modern graphics hardware. This allowed us to devise an OpenGL application that implements the full algorithmic pipeline of our techniques. Besides increasing the performance, this approach offers direct access to the z-buffer data which can be directly used for further processing. In this way, we avoid the overhead of reading the depth data back to the CPU or writing it to a file as input for another application. In our implementation all intermediate results are stored as floating-point textures, such that the whole pipeline remains on the GPU. This results in a single application that achieves real time performance and allows the user to seamlessly control meaningful parameters as well as transform objects in the scene. As a direct result of this performance boost, bas-reliefs of animated scenes can be generated with our method.

5.2 User Interface

We build a graphical user interface on top of the hardware implementation of both presented algorithms which allows setting the input parameters in meaningful ranges with the help of sliders. We provide a viewing interface with two different modes. In the first mode, an artist can import a mesh or a height field and switch between orthogonal or perspective projection. The scene can then either be transformed until the desired pose is reached or it can be constantly rotated around an arbitrary axis. The second mode displays the acquired new height field after the bas-relief computation of the current view of the input, achieved by the selected method. The effect of changing parameters is visible without delay. In this way, the whole process of bas-relief generation becomes truly interactive. We can export the generated bas-relief as a 2D height matrix or in the form of a 3D mesh. Fig. 5 shows several screen shots of our user interface.

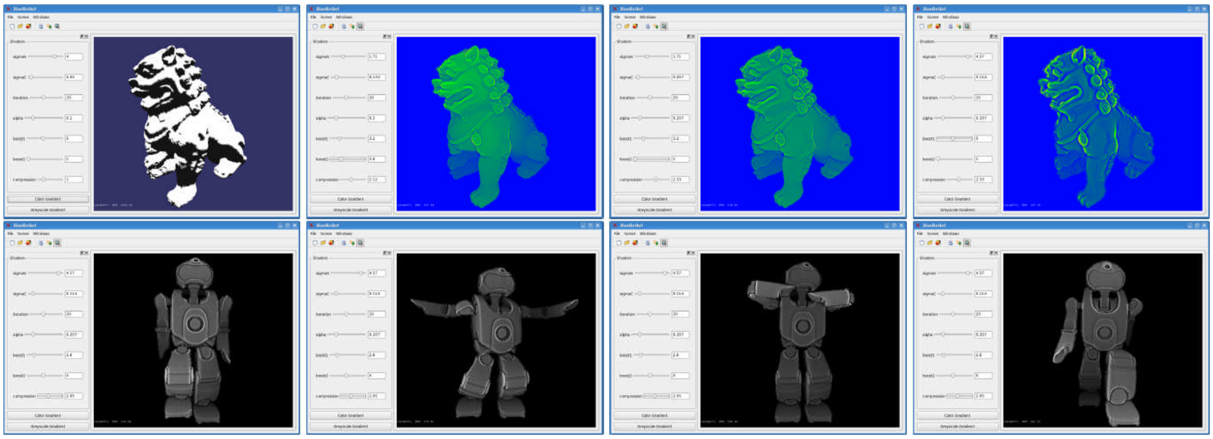

Fig. 5: (top) The Chinese Lion-dog model in our viewer, followed by three color coded bas-reliefs (green to red) achieved with three different parameter settings; (bottom) four final height maps in gray scale of a dancing robot sequence; in both cases the displayed results were produced with the range domain approach.

5.3 Performance

For the gradient method, we considered it best to include NVidia Cuda 2.0 as it allows for an elegant solution of the Poisson equation via fast Fourier transformation. The transfer from OpenGL textures into Cuda space is the main bottleneck of the pipeline, as it consumes more than 70% of the overall computation time. We are confident that this problem will be addressed in future hardware. The computation of the average values and standard deviation for outlier removal are achieved via mip-mapping. Given the current limitations of graphics hardware, we can only achieve single precision. Moreover, as we rely on textures we need to consider the available graphics memory as a limited resource. At the moment, this restricts us to reliefs of a resolution up to 2048x2048.

In any case, the performance is independent of the scene complexity. In the range domain approach it depends on the number of Laplace iterations, but since we need fewer textures and no texture transfer is necessary, the range domain pipeline is much faster and can handle higher resolutions.

For scenes with a resolution of 1024x1024 pixels the gradient domain approach achieves 25 FPS whereas the computations in the range domain reach 180 FPS when 20 Laplacian iterations are applied. The performance was measured using an NVidia Geforce8800 Ultra with 800MB of memory in an AMD 2,66GHz Dual Opteron with 3GB RAM and PCI Express.

As we rely on bilateral filtering in both approaches, the overall performance could be further enhanced by using bilateral filters that work with fast Gauss transform, as for instance the variant presented in [1], for which the authors publicly provided the Cuda code recently. Our initial tests suggest an improvement of the runtime by up to 8%.

6 RESULTS AND DISCUSSION

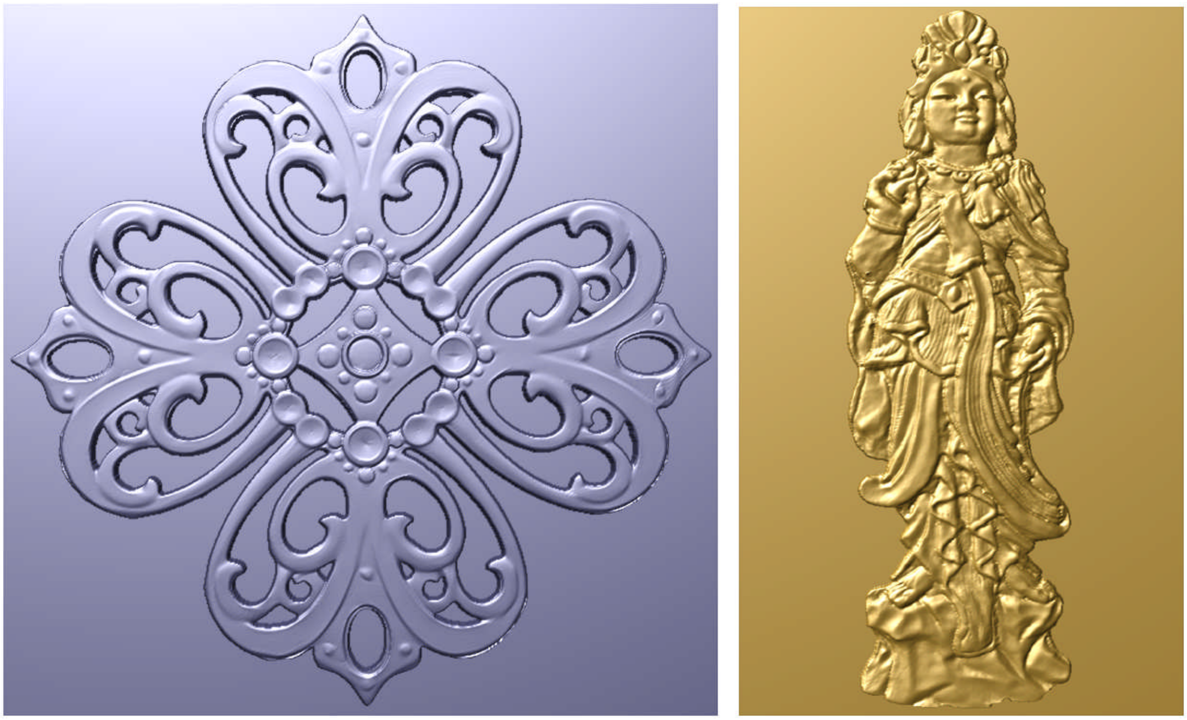

Fig. 6: An ornament bas relief (left) generated with the range domain method. The bas relief of the Malaysia Statue (right) is generated with the gradient domain approach.

For visualization purposes, the bas-reliefs are represented as triangle meshes that contain the same number of vertices as there are pixels in the depth map. Each vertex is displaced in the z-direction according to its entry in the bas-relief height field. In all of our experiments, the models were compressed in a way that the depth range is equal to 1.5% of their largest dimension (x or y). All results presented here contain a slightly elevated silhouette mainly due to the final normalization step. Figures 2, 3 and 6 show typical results of our approach. The range domain method emphasizes the step between the armadillo's jaw and the breast (Fig. 2 left). The muscles and the fangs are well reproduced and the overlapping parts of the ears support the plastic impression. The example of the ornament (Fig. 6 left) illustrates that this algorithm is able to preserve the correct depth order. The third example shows the socle of a Thai statue (Fig. 3 left) compressed with the gradient domain technique. Sharp small features and high frequency details like the dragon scales and the teeth are well preserved.

The cavities on the proboscis of the elephant as well as the relief on the pillar inside of the model are faithfully reproduced. The bas-relief of the Malaysian statue (Fig. 6 right) correctly carries the facial expression as well as detailed geometric features such as the jewelry and garment wrinkles. Inspection of the color coded height maps before and after the compression reveals that our processing steps are well justified and contribute effectively towards achieving the desired design goals.

We were kindly provided with two reliefs created by the authors of [14] which we use for comparison. As we do not have access to a current implementation of their work, we tried our best to reproduce the same pose and scene conditions to ensure a fair comparison. Fig. 7 compares our results to the ones from [14]. On both models, their results convey a more plastic and life-like impression. For the castle it becomes obvious that the range domain approach produces the better result in comparison with the gradient method. This is due to the fact that the model does not exhibit much high frequency structures but instead consists of large steps which are better reproduced because they are covered in the base layer as illustrated by the crosses on the entrance roof. In contrast, the eye-ball of the dragon is reproduced more faithfully in the gradient domain method and does not appear as flat as it does in the range domain result.

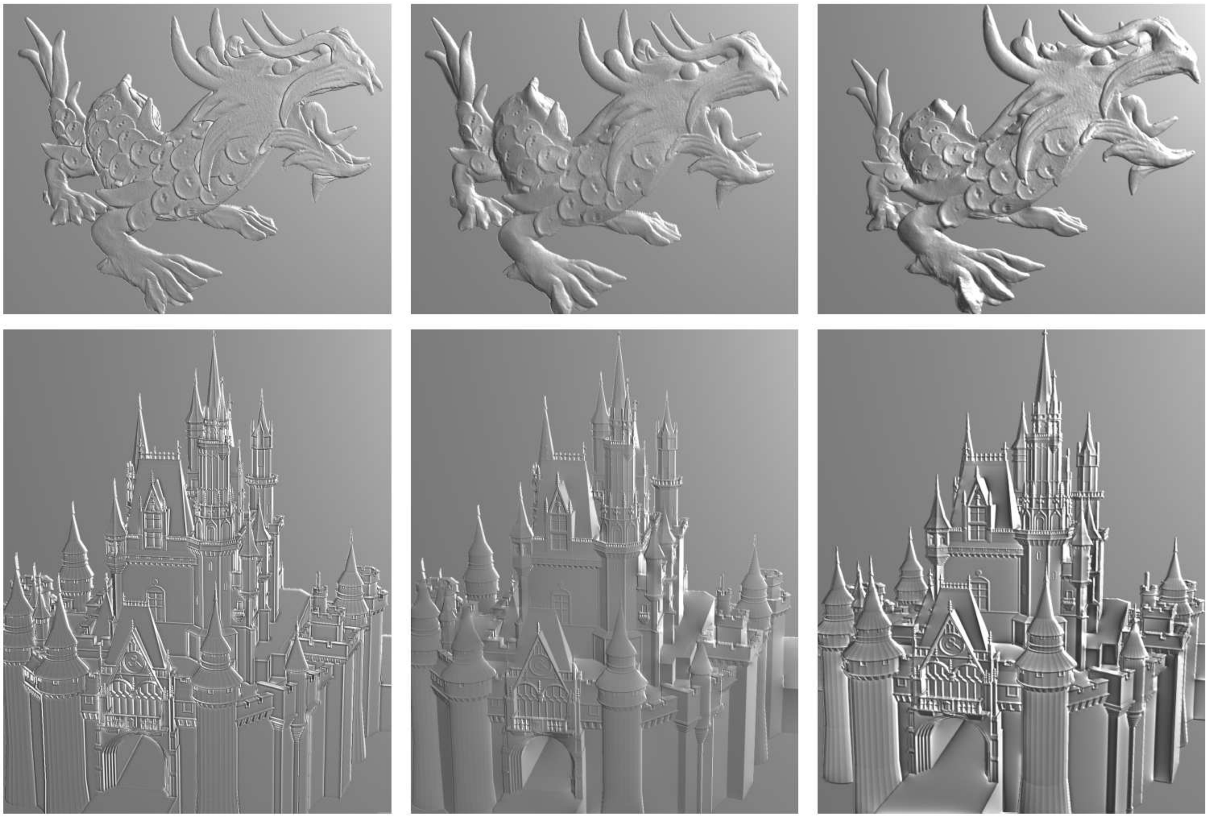

Fig. 7: (left) outcomes of the range domain method, (middle) bas-reliefs achieved with the gradient approach, (right) results of the technique of Weyrich et al.

The range domain method offers three layers which can be re-weighted; it is very fast and produces good results especially when coarser structures should be preserved. Keeping large features leads to a more plastic impression but wastes a lot of depth and impairs the visibility of small details when higher compression ratios should be achieved. The gradient technique only allows re-weighting of two different frequency bands, it is slightly slower but requires less user intervention and it keeps a sharp and faithful impression of small details for an arbitrary compression ratio since coarser features only play a minor role. Finally it also allows the production of seamless collages and cubism like bas-reliefs. So the decision between the two techniques should be driven by the constitution of the scene and the desired depth range. The approach from [14] gives the opportunity to use an arbitrary number of frequency levels. This allows distinguishing between very small details and noise. This level of control, however, comes at the price of long computation times and the necessity for laborious re-iterations during design.

7 CONCLUSION

We presented two real-time semi-automatic CAD tools intended to support professional artists and enthusiasts in creating bas-relief artworks from virtual scenes. The tools are capable of producing perceptually plausible digital bas-reliefs. In addition, we demonstrated how to use our approach for generating cubism like bas-relief scenes. The whole framework is intuitive and easy to implement and use.

ACKNOWLEDGMENTS

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. We thank Kun Xu and the Graphics and Geometric Computing Group at Tsinghua University for providing the robot 3D animation sequence.

REFERENCES

[1] Adams, A.; Gelfand, N.; Dolson, J.; Levoy, M.: Gaussian KD-trees for fast high-dimensional filtering, ACM Transactions on Graphics, 28(3), 2009, 1-12.

[2] Alldrin, N.; Mallick, S.; Kriegman, D.: Resolving the Generalized Bas-Relief Ambiguity by Entropy Minimization, IEEE Conf. on Computer Vision and Pattern Recognition, 2007.

[3] Belhumeur, P. N.; Kriegman, D. J.; Yuille, A. L.: The Bas-Relief Ambiguity, International Journal of Computer Vision, 35(1), 1999, 33-44.

[4] Cignoni, P.; Montani, C.; Scopigno, R.: Automatic Generation of Bas- and High-Reliefs, Journal of Graphics Tools, 2(3), 1997, 15-28.

[5] Debevec, P.; Reinhard, E.: High-Dynamic-Range Imaging: Theory and Applications, SIGGRAPH 2006 Course #5, http://www.siggraph.org/s2006/main.php?f=conference&p=courses&s=5

[6] Kerber, J.; Belyaev, A.; Seidel, H.-P.: Feature Preserving Depth Compression of Range Images, Proceedings of the 23rd Spring Conference on Computer Graphics, 2007, 110-114.

[7] Kerber, J.; Tevs, A.; Zayer, R.; Belyaev, A.; Seidel H.-P.: Feature Sensitive Bas Relief Generation, IEEE International Conference on Shape Modeling and Applications Proceedings, 2009, 148-154.

[8] Paris, S.; Kornprobst, P.; Tumblin, J.; Durand, F.: A Gentle Introduction to Bilateral Filtering and its Applications, IEEE CVPR 2008 tutorial, http://people.csail.mit.edu/sparis/bf_course/

[9] Pasko, A. A.; Savchenko, V.; Sourin, A.: Synthetic Carving Using Implicit Surface Primitives, Computer-Aided Design, 33(5), 2001, 379-388.

[10] Policarpo, F.; Oliveira, M. M.; Comba, J. L. D.: Real-time relief mapping on arbitrary polygonal surfaces, ACM SIGGRAPH 2005, 2005, 155-162.

[11] Song, W.; Belyaev, A.; Seidel, H.-P.: Automatic Generation of Bas-reliefs from 3D Shapes, Proceedings of the IEEE International Conference on Shape Modeling and Applications, 2007, 211-214.

[12] Sun, X.; Rosin, P. L.; Martin, R. R.; Langbein, F. C.: Bas-Relief Generation Using Adaptive Histogram Equalisation, IEEE Transactions on Visualization and Computer Graphics, 15(4), 2009, 642-653.

[13] Tan, P.; Mallick, S. P.; Quan, L.; Kriegman, D.; Zickler, T.: Isotropy, Reciprocity and the Generalized Bas-Relief Ambiguity, IEEE Conf. on Computer Vision and Pattern Recognition, 2007.

[14] Weyrich, T.; Deng, J.; Barnes, C.; Rusinkiewicz, S.; Finkelstein A.: Digital bas-relief from 3D scenes, ACM Transactions on Graphics, 26(3), 2007.