EUROGRAPHICS 2015 / O. Sorkine-Hornung and M. Wimmer
(Guest Editors)

Volume 34 (2015), Number 2

High Reliefs from 3D Scenes

Sami Arpa, Sabine Süsstrunk, and Roger D. Hersch

School of Computer and Communication Sciences, EPFL, Switzerland

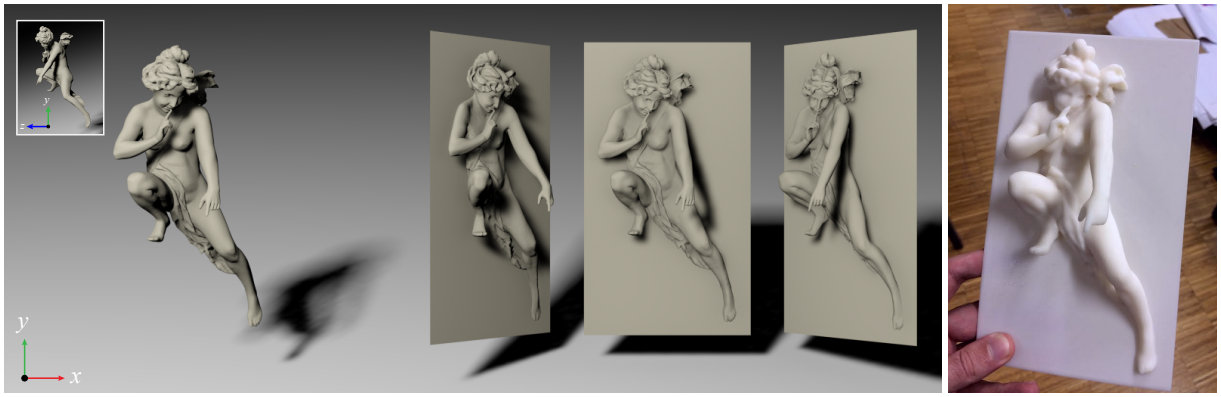


Figure 1: *With an original 3D scene* (left)*, we generate a high relief* (middle) *that incorporates free-standing sculpture elements located within a limited depth range. Today, high reliefs can be produced thanks to additive fabrication technologies such as 3D printing* (right).

Abstract

We present a method for synthesizing high reliefs, a sculpting technique that attaches 3D objects onto a 2D surface within a limited depth range. The main challenges are the preservation of distinct scene parts by preserving depth discontinuities, the fine details of the shape, and the overall continuity of the scene. Bas relief depth compression methods such as gradient compression and depth range compression are not applicable for high relief production. Instead, our method is based on differential coordinates to bring scene elements to the relief plane while preserving depth discontinuities and surface details of the scene. We select a user-defined number of attenuation points within the scene, attenuate these points towards the relief plane and recompute the positions of all scene elements by preserving the differential coordinates. Finally, if the desired depth range is not achieved we apply a range compression. High relief synthesis is semi-automatic and can be controlled by user-defined parameters to adjust the depth range, as well as the placement of the scene elements with respect to the relief plane.

1. Introduction

High relief, also referred to as alto-rilievo, is a sculpting technique in the plastic arts [MM72]. In contrast to bas reliefs, in which the scene elements are projected into a very narrow depth range, high reliefs contain elements that are detached from the relief plane (Fig. 2). This makes the design and production of high reliefs more complicated than the creation of bas reliefs. Preserving 3D shape characteristics in a limited depth range is the main challenge.

The production techniques used for bas reliefs such as milling and carving, in which the shape is created by removing material from the main mass, are not easily applicable to high reliefs. However, recent advances in additive manufacturing such as 3D printing simplify the production of high reliefs. These additive technologies enable building the parts that are detached from the plane using special support materials that can be removed after the print. The new materials available for 3D printers, such as thermoplastics and

photopolymers, have a transparent, matte, or shiny appearance. They thus enable creating high reliefs for decorative or artistic purposes.

Figure 2: *High relief examples.* Left: *Greek relief in marble, Girl with Doves, 300 B.C.;* Middle: *Hero-City Obelisk, St. Petersburg, 1985;* Right: *Cherub sculpture in high relief.*

While techniques have lately been proposed for the design of bas reliefs [KWC*12], there have not been many attempts to address the challenges of high reliefs. The fundamental challenges for high relief design are the following [Hof39]:

- **Depth discontinuities.** Parts of the scene are detached from the relief plane and result in depth discontinuities (Fig. 3). While depth discontinuities are avoided in bas reliefs, they are emphasized in high reliefs. The resulting strong shadow effects enhance the illusion of depth. Depth discontinuities enable viewing the high relief differently from different viewing orientations. This creates the illusion of seeing a real 3D scene. However, preserving both the original shape properties and the depth discontinuities within a limited depth range is challenging.
- **Surface continuity.** The continuity between the "high" and the "low" parts of the relief needs to be ensured. The high parts preserve the depth to a large extent and the low parts are very close to the plane and look like bas reliefs.
- **Perspective deformation.** The depth illusion can be further enhanced by applying a perspective deformation to the 3D scene.

The method we present for designing high reliefs of given 3D scenes addresses these challenges and assists both amateur and professional users. In order to enhance the depth perception of the observer, we first apply a perspective deformation of the scene according to a given vanishing point. Then we bring certain scene elements to the relief plane and calculate all other elements by using differential coordinates [Ale03, KG00]. The continuity and fine details of the original scene and the depth discontinuities of the high surface parts are preserved (Fig. 1). Our method allows the user to control the depth range, as well as the parts of the scene that are brought to the relief plane.

2. Background and Related Work

Automatic generation of reliefs has recently become a significant research topic. However, most works have focused on the design techniques for the creation of bas reliefs [KWC*12].

The dominating approach for designing a bas relief is based on a height field as initially presented in the work of Cignoni et al. [CMS97]. The input scene is represented as a simple height field. The range of this height field is then reduced to generate plausible bas reliefs.

The bas relief synthesizing methods mainly addressed the trade-off between depth compression and the preservation of the details of the original 3D scene. Cignoni et al. [CMS97] apply a non-linear compression directly on the points of the height field. The compression is small for the points that are close to the viewpoint and large for those that are far. Although this method produces acceptable bas reliefs for small compression ratios, it fails to preserve the details for scene elements having a wide range of depth. To solve this problem, several works [SBS07, WDB*07, KTB*09] try to preserve most of the details from the scene by operating in the gradient domain [FLW02] instead of the depth map. Additionally, Weyrich et al. [WDB*07] apply a multiscale approach for gradient compression. Some recent methods [SRML09, BH11] work both on the depth range and in the gradient domain to further enhance the resulting bas reliefs.

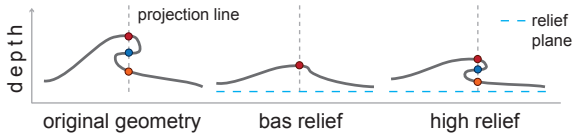


Figure 3: *Bas and high relief representation in 2D.*

Since state-of-the-art solutions for bas reliefs [KWC*12] are based on a height field, they are not adequate for generating high reliefs. As shown in Fig. 3, a bas relief can represent only a single point projected from the scene. In contrast, high reliefs may contain multiple points located on the same projection line. For the creation of high reliefs, Cignoni et al. [CMS97] propose to first decompose the scene into a near and far region, where the far region is compressed with a bas relief synthesizing algorithm and the near region is simply stitched to the compressed far region. This method yields acceptable high reliefs for several types of scenes where the near and far regions are discontinuous. However, it is not applicable for the most of complex 3D scenes. The authors also do not address the problem of mapping the high relief part of the scene into a limited depth range.

A recent work by Schuller et al. [SPSH14] introduces *appearance-mimicking surfaces* as a generalization of bas relief synthesis for arbitrary target surfaces. This method does not require a height field as input data but uses the 3D scene geometry. Given a number of constraints such as target shapes, viewpoints, and space restrictions, they produce an optimal relief providing the desired appearance from a

certain viewpoint. In contrast, our approach aims at creating high reliefs that can be viewed from a wide range of viewing angles with an appearance similar to the original 3D shape.

In order to construct visually pleasing and continuous high relief surfaces we use differential coordinates. Representing 3D surfaces in differential coordinates, also known as Laplacian coordinates, has been previously proposed [Ale03, KG00]. In this representation, each vertex is represented by its difference from its topological neighbours. Laplacian coordinates can be efficiently used to re-build a surface by solving a sparse linear system [SCOL*04].

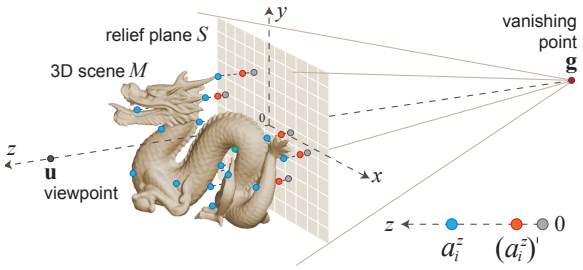


Figure 4: *A scene M with viewpoint* **u**, *relief plane S, vanishing point* **g**, *and* $k = 16$ *attenuation points.*

3. High Reliefs

We propose a high relief synthesis technique that is significantly different from the range compression methods used to produce bas reliefs. We first select a number of relevant scene points as attenuation points and move these points towards a relief plane by considering the local relationships of those points with each other. Then we reconstruct the surface by using the new positions of the attenuation points and the differential coordinates of the original scene. This brings the scene elements close to the relief plane while maintaining the continuity and fine details of the surface of the original scene. Furthermore, high reliefs keep the presence of separate scene elements at different depth levels. Depth discontinuities are therefore preserved.

Let us introduce the notations used hereinafter. We consider a relief plane $S \in R^3$ and a 3D scene $M = \{V, \Delta V\}$, where $V = \{\mathbf{v}_1, ..., \mathbf{v}_n\} \subset R^3$ incorporates the Cartesian coordinates $\mathbf{v}_i$ of n vertices. Each vertex $\mathbf{v}_i \in V$ has the coordinates $\mathbf{v}_i = [v^x\ v^y\ v^z]$. The differential coordinates $\Delta V = \{\Delta\mathbf{v}_1, ..., \Delta\mathbf{v}_n\} \subset R^3$ comprise, for each vertex, the weighted differences from their neighbours: $\Delta\mathbf{v}_i = [\Delta v^x\ \Delta v^y\ \Delta v^z]$. The scene M is further defined by the user controlled viewpoint **u** and the vanishing point **g**, which guide the perspective deformation. We partition the scene into k clusters $C = \{c_1, ..., c_k\}$. In each cluster c_i, we select one attenuation point a_i that we use to control the proximity of the scene elements to the relief plane. Fig. 4 shows the scene, the viewpoint, the relief plane, and the vanishing point within a coordinate system centered in the relief plane.

3.1. Forced perspective

Perspective foreshortening [DH04] is a key tool to achieve a depth illusion when projecting 3D scenes onto a 2D plane. However, it is not applicable for high reliefs since we do not project 3D points onto a 2D plane. Nevertheless, since we limit the depth range, deforming the geometry along vanishing lines is useful to enhance the perception of depth. This type of perspective deformation is applied in several architecture and sculpting works. For instance, the Borromini corridor in Palazzo Spada is perceived to be much longer than its actual length. This is due to the parallel lines of the corridor that are deformed so as to converge to a vanishing point [Arn74, p. 273]. In a similar manner, we create a forced perspective on the 3D scene by applying a one point perspective deformation, see Appendix II. The deformation is controlled with the vanishing point $\mathbf{g} = [g^x\ g^y\ g^z]$ located behind the relief plane and the viewpoint **u**. Choosing **g** at infinity ($g^z = \infty$) does not cause any deformation of M, while choosing it closer to the relief plane S creates a corresponding perspective deformation (Fig. 5). Note that, since we limit the perspective to one vanishing point, non-linear deformations occur on the lines that are neither parallel nor perpendicular to the viewing direction. In order to achieve visually interesting forced perspective effects, and to avoid non-linear deformations we recommend transforming the scene so that dominant lines become parallel or perpendicular to the viewing direction. Alternatively, choosing the vanishing point further away from the relief plane decreases the non-linearities but reduces the forced perspective effect.

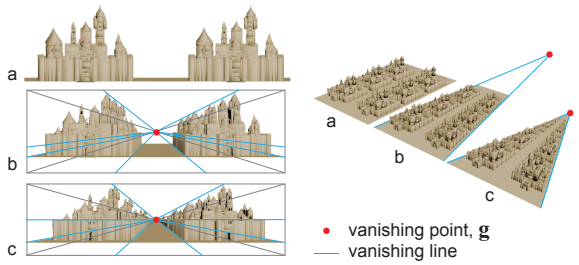


Figure 5: *Perspective deformation.* a: *Original input scene;* b: *the scene deformed for a far vanishing point;* c: *the scene deformed for a near vanishing point.*

3.2. Bring it to the plane

Instead of directly compressing the depth range, we bring scene elements towards the relief plane S and attach some of them to the plane. At the same time we aim to preserve the overall shape. We achieve this by selecting several scene vertices as attenuation points. These attenuation points are then migrated towards the relief plane S. The positions of all other vertices are calculated according to the new positions of the attenuation points and the differential coordinates ΔV of the vertices.

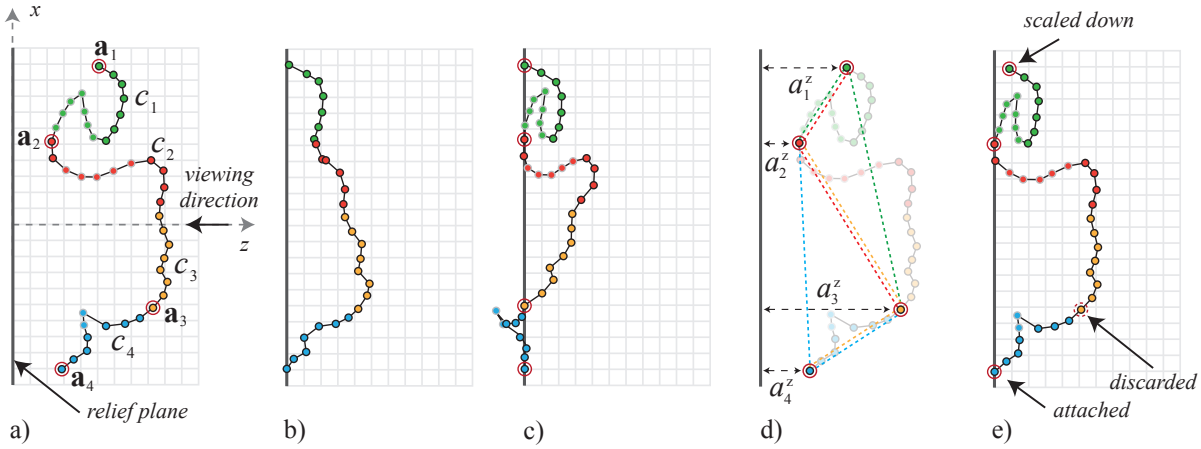


Figure 6: *Clustering and selection of attenuation points on a simple curve in 2D. Each circle indicates a vertex. Vertices with dark borders are visible vertices and vertices with light borders are invisible vertices. The vertices having the same color belong to the same cluster. a) The vertices are clustered according to their x positions and the visible vertex closest to the relief plane is selected as attenuation point in each cluster. b) The result of reconstructing the scene as bas relief using the method of Weyrich et al. [WDB*07]. c) The reconstruction of the scene if all attenuation points are attached to the relief plane. d) The differential values for attenuation points are calculated by taking the differences from their 2 nearest neighbours, illustrated with the respective color lines. In 3D, we use 4 nearest neighbours. e) The reconstructed scene after moving the attenuation points considering their differential values.*

Attenuation points. The selection of attenuation points determines to a large extent the final high relief geometry. Our goal is to select those attenuation points as control points that enable bringing the scene towards the relief plane by satisfying the following requirements:

- Ensure that all vertices that are visible from viewpoint $\mathbf{u}$ stay in front of the relief plane.
- Use the least number of attenuation points k in order to avoid excessive deformations.
- Preserve the original depth order of all scene vertices in local neighbourhoods.

In order to meet these criteria, we apply the following procedure. A spatially well distributed selection of attenuation points is required to control the scene elements. Therefore, we partition the scene into spatial clusters C and select one vertex from each cluster c_i as attenuation point $\mathbf{a}_i$ that becomes its control point. We apply a k-means spatial clustering [M*67] to the scene vertices V using only their x and y coordinates as cluster data and k as the desired number of clusters. With this strategy, we do not partition the scene in the z direction. This prevents self intersection of scene parts when the attenuation points are moved towards the relief plane.

In each cluster, we select as attenuation point $\mathbf{a}_i \in V$ the vertex visible from viewpoint $\mathbf{u}$ having the smallest Euclidean distance to the relief plane S. This vertex may become an attachment point to the relief plane. This procedure provides well distributed attenuation points that constraint visible vertices to stay in front of the relief plane. This can be shown more easily in 2D than in 3D. Fig. 6a shows a scene comprising a curve in 2D, its clusters and their corresponding attenuation points.

Next we scale down the distance of each attenuation point $\mathbf{a}_i = [a_i^x\ a_i^y\ a_i^z]$, $\mathbf{a}_i \in c_i$ to the relief plane. Attaching all attenuation points to the relief plane, as shown in Fig. 6c, creates problems such as undesired deformations and the migration of visible parts of the scene behind the relief plane when the surface is reconstructed (see cluster c_4 in Fig. 6c). Such deformations appear when an attenuation point is more distant to the relief plane than its neighbouring attenuation points. In order to prevent this, we develop a strategy where attenuation points are either discarded, scaled down, or attached to the relief plane. For this purpose, we define the differential relief distance of an attenuation point as the difference between its relief distance and the average relief distances of its neighbours:

$$\Delta a_i^z = a_i^z - \frac{1}{|N|} \sum_{j \in N} a_j^z \qquad (1)$$

where N is the set of 4 nearest neighbours of the attenuation point $\mathbf{a}_i$. The nearest neighbours are searched by calculating the Euclidean distances in x, y coordinates between an attenuation point $\mathbf{a}_i$ and other attenuation points. Then, we map all differential relief distances to the interval between 0 and 1:

$$\rho_i = \frac{\Delta a_i^z - \Delta a_{min}^z}{\Delta a_{max}^z - \Delta a_{min}^z} \tag{2}$$

where Δa_{min}^z is the minimal differential relief distance and Δa_{max}^z is the maximal differential relief distance. Depending on the value of the relative differential relief distance ρ_i, of an attenuation point a_i, we either attach it to the relief plane, scale it down, or discard it:

$$\begin{aligned} \rho_i > t_1 &\quad\rightarrow\quad \text{discard} \\ t_2 \leq \rho_i \leq t_1 &\quad\rightarrow\quad \text{scale down} \\ \rho_i < t_2 &\quad\rightarrow\quad \text{attach} \end{aligned}$$

where t_1 is the threshold for discarding an attenuation point and t_2 is the attachment threshold. In our experiments, we heuristically chose $t_1 = 0.6$ and $t_2 = 0.3$, respectively. These values give good results in all our examples.

A high ρ_i, which corresponds to a high positive differential relief distance, indicates that the attenuation point is further away from the relief plane compared to its neighbours. It is not a suitable attenuation point since it might cause excessive deformations (see $\mathbf{a}_3'$ and c_4 in Fig. 6c). In contrast, a small ρ_i, which corresponds to a low negative differential relief distance, indicates that the corresponding attenuation point is closer to the relief plane compared to its neighbours and that it can be an attached to the relief plane (see $\mathbf{a}_2'$ and $\mathbf{a}_4'$ in Fig. 6d). For attenuation points with $t_2 \leq \rho_i \leq t_1$, we reduce their relief distances a^z. In order to scale down large absolute differentials more, we scale differentials Δa_i^z by using an attenuation function proposed by Bian et al. [BH11].

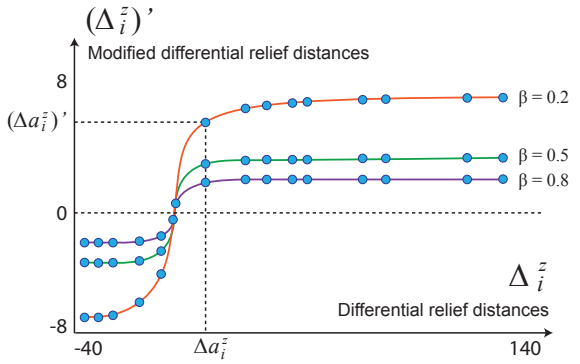


Figure 7: *Modified differential relief distances for $k = 16$ attenuation points after applying the attenuation function proposed by Bian et al. [BH11] (see Eq. 3) for the dragon model in Fig. 4.*

$$(\Delta a_i^z)' = \frac{\arctan \beta_i \Delta a_i^z}{\beta_i} \tag{3}$$

We use attenuation factor $\beta_i = 0.2$ for all attenuation points if not stated otherwise. As shown in Fig. 7, it affects large differantial values very strongly. A wide range of differential values is almost equalized [KWC*12].

Next we find the new relief distances of attenuation points by reconstructing them using their modified differential relief distances. The attenuation points selected as attachment points represent positional constraints:

$$(a_i^z)' = 0, \quad i \in \{b, ..., k\}, \; b < k \tag{4}$$

where $\{b, .., k\}$ are the indices of the attachment points. Under these constraints, we find the distances of the other attenuation points $(a_i^z)', \quad i \in \{1, ..., b-1\}$ using the modified differentials. However, instead of directly fixing attachment points, we recalculate them in a least square sense as recommended by Sorkine et al. [SCOL*04] by minimizing the following cost function:

$$E((a_1^z)', ..., (a_k^z)') = \sum_{i=1}^{k} \|(\Delta a_i^z)' - \Delta(a_i^z)'\|^2 + \sum_{i=b}^{k} \|(a_i^z)'\|^2 \tag{5}$$

The first term in the optimization minimizes the square differences between modified differential relief distances $(\Delta a_i^z)'$ and the differential values of the newly obtained relief distances $\Delta(a_i^z)'$. The second term minimizes the square distances to the relief plane for those attenuation points that are selected as attachment points. This optimization problem is solved as a sparse linear systems of equations [SCOL*04] (see Appendix IV). Fig. 6e shows the result for a curve in 2D.

Construction. We construct the final relief surface by using the new positions of the attenuation points $\mathbf{a}_i' = [a_i^x \; a_i^y \; (a_i^z)']$ as positional constraints. As discussed previously, the differential coordinates ΔV enable preserving the local characteristics of the scene while moving scene elements to the plane. We calculate the differential coordinates $\Delta \mathbf{v}_i$ of a vertex $\mathbf{v}_i$ by using the cotangent weights as described in Sorkine [Sor05] (see Appendix III). Next we reconstruct the surface by fixing the new positions of the attenuation points:

$$\mathbf{v}_i' = \mathbf{a}_i', \quad i \in \{m, ..., n\}, \; m < n \tag{6}$$

where $\{m, .., n\}$ are the k indices of the vertices that are selected as attenuation points. Under these constraints, we find the new positions of all other vertices $\mathbf{v}_i', \quad i \in \{1, ..., m-1\}$ using the differential values $\Delta \mathbf{v}_i', \quad i \in \{1, ..., m-1\}$. We minimize the square differences between the differential coordinates of new vertex positions $\Delta \mathbf{v}_i'$ and original differen-

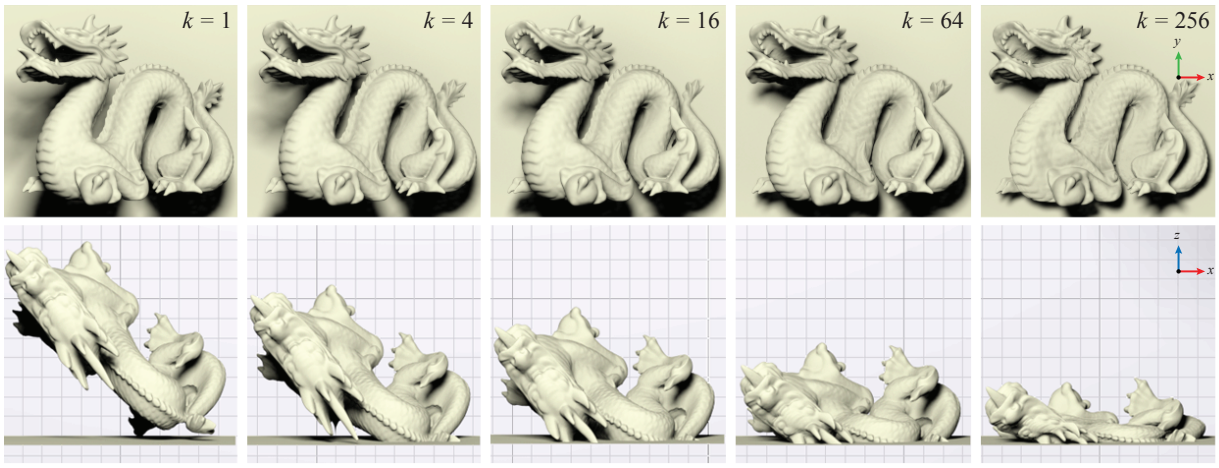


Figure 8: *Increasing the number k of attenuation points increases the number of scene elements that are brought to the relief plane and decreases the overall depth range.*

tial coordinates $\Delta\mathbf{v}_i$ for all vertices except the attenuation points.

$$E(V') = \sum_{i=1}^{m-1} \| \Delta\mathbf{v}'_i - \Delta\mathbf{v}_i \|^2 \tag{7}$$

This optimization problem is solved as a sparse linear systems of equations (see Appendix V). Instead of treating the constraints in the least-squares sense as suggested by Sorkine et al. [SCOL*04], we substitute them into the linear system.

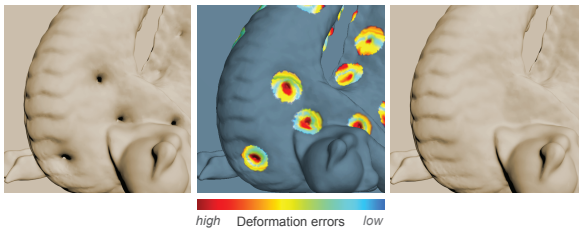


Figure 9: *Undesired deformations become more visible as the number k of attenuation points is increased* (left). *We detect local deformation errors by comparing the depth orders* (middle) *and apply a reconstruction to correct the errors* (right).

Restoration. Increasing the number k of attenuation points increases the number of scene elements that are brought to the relief plane and decreases the overall depth range (Fig. 8). However, having sparsely distributed positional constraints results in undesired deformation errors. Since we apply a least square solution, errors are locally distributed around the attenuation points (Fig. 9). The usage of higher order differential coordinates or the relaxation of positional constraints in the construction step can smooth the errors, but result in unpleasant global changes, see the supplementary material. Instead, we detect the errors and correct them by applying a local reconstruction procedure using higher order differential coordinates. First, we detect local errors by comparing the depth order of the current vertices V' with the depth order of the original vertices V. For each attenuation point $\mathbf{a}_i$, we check all vertices within a certain radius (i.e. half of the largest distance between two points in the current cluster considering the original vertex positions V). Depth order error ε is calculated for a vertex $\mathbf{v}$ in the neighbourhood of $\mathbf{a}_i$:

$$\varepsilon = \frac{| v'_o - v_o |}{p} \tag{8}$$

where p is the number of vertices in the neighbourhood of $\mathbf{a}_i$. v'_o and v_o denote the current and original depth order of vertex $\mathbf{v}$, respectively. The depth order is obtained by presorting all p vertices according to their distances to the relief plane. In this manner, we calculate errors for all vertices which are around attenuation points and assign $\varepsilon = 0$ for all other vertices. Next we reconstruct the surface parts having high errors. All vertices with error $\varepsilon < t_3$ are fixed as positional constraints. We use $t_3 = 0.2$. This value has been chosen by carrying out extensive tests. To avoid discontinuities, we use higher order differential coordinates in this step, as described in Appendix VI. As shown in Fig. 9, with this strategy, we remove local errors without relaxing the depth range or introducing new deformations.

As a final step we eliminate all vertices that are displaced behind the relief plane S.

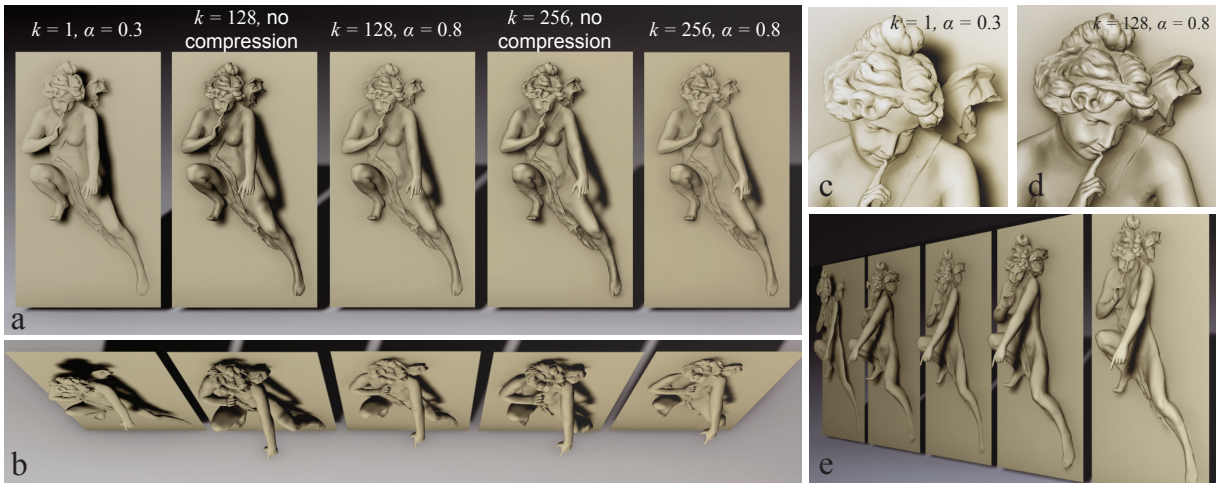


Figure 11: *Our results preserve to a large extent the fine details and the properties of the original scene within a limited depth range. With a simple depth compression method equivalent to the prior art depth range compression strategies* (a: $k = 1$, $\alpha = 0.3$; c: close-up)*, the details are lost and the scene is not well attached to the relief plane* (b)*. With more attenuation points* (a: $k = 128$, no compression; d: close-up)*, the scene elements are brought to the relief plane without inducing a loss of details.*

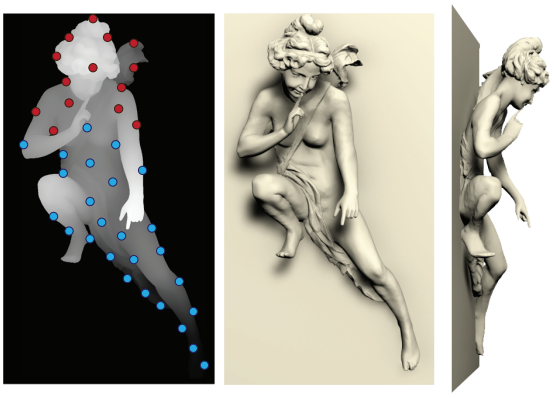


Figure 10: *The users can control the attenuation factor β_i for each attenuation point and possibly release attachment points. The red dots indicate the attenuation points with $\beta_i = 0.1$ and blue ones with $\beta_i = 0.8$. Smaller β_i values enable scene elements to be detached from the relief plane.*

3.3. Compression and artistic control

Although our construction method limits the depth range according to the number k of attenuation points, we also apply a range compression method if the desired depth range is very limited. Increasing k decreases the depth range as shown in Fig. 8, but more undesired deformations may occur. In these cases, a range compression method may be preferred for further compression. We apply a range compression similar to the method of Cignoni et al. [CMS97], where the vertices closer to the relief plane are compressed more than the far ones. We automatically assign a compression ratio for each vertex depending on a user given maximal α_{max} and minimal α_{min} compression ratio. A local compression value α_i is calculated for each vertex v_i by mapping the range of relief distances $(v_i^z)'$ in the interval between α_{min} and α_{max}. Finally, the relief distance $(v_i^z)'$ of a vertex is multiplied by its local compression ratio α_i. For the examples in the paper, we only indicate α_{max} by referring to it as α and use $\alpha_{min} = \alpha_{max}/2$.

As shown in Fig. 11 and Fig. 13 as well as in the supplementary material, we obtain different results depending on the selection of k, α, β, and vanishing point $\mathbf{g}$. For the artistic control of the resulting high relief, these parameters can be set by the designers. Furthermore, they can also define the attenuation factors β_i for the attenuation points, see Eq. 3. Changing β_i globally enables determining how much the scene sinks into the relief plane. Furthermore, changing it locally for each attenuation point enables tuning how the different parts of the high relief surfaces stand out from the relief plane (Fig. 10).

4. Results

We discuss the results and illustrate potential usages based on different settings. As discussed in Sec. 3, we provide artistic control for the number of attenuation points k, the depth compression value α, as well as the attenuation factor β, possibly adapted for each attenuation point. Having parts of the scene detached from the relief plane enhances the depth perception to a great extent. In contrast to bas reliefs, the preservation of depth discontinuities yields strong shadows. Fig. 11 shows a comparison between a simple compression of the depth range ($k = 1$, $\alpha = 0.3$) and our method

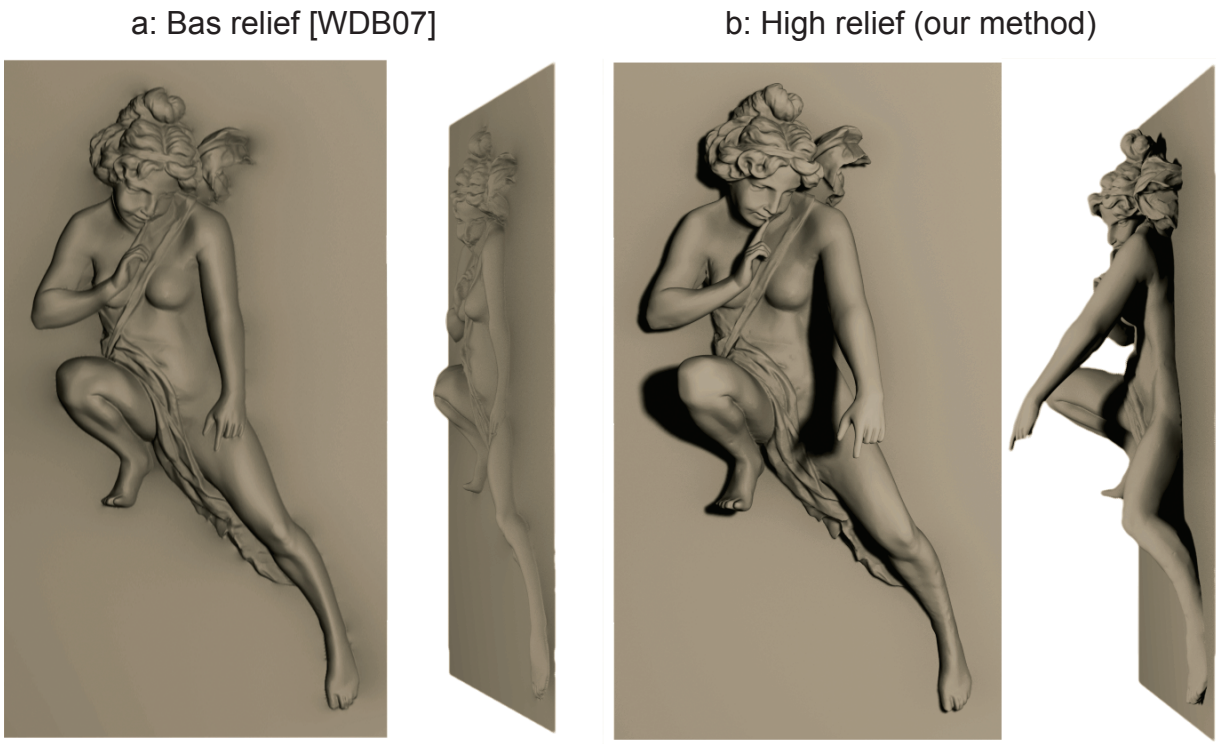


Figure 12: *The comparison of bas relief and high relief at different angles under the same lighting conditions.*

with different numbers of attenuation points and range compression values. Our strategy preserves the details, ensures surface continuity, and maintains depth discontinuities of the scene. A simple depth compression strategy fails to preserve the scene details and the overall scene is not well attached to the relief plane. Increasing the number k of attenuation points decreases the depth range but at the same time introduces deformations of the scene geometry. If a further decrease in depth range is desired, the compression ratio α enables a range compression. If no deformation is tolerated and a limited depth range is desired, we observe that selecting $k = 128$ and $\alpha = 0.8$ gives the best results for many scenes. Fig. 13 and the supplementary material show additional results for different types of scenes.

Choice of methods. We use different solutions for finding the new positions of the attenuation points (Eq. 5) and of all vertices (Eq. 7). As shown by Sorkine et al. [SCOL*04], using the least-squared method is preferable in most of scenarios. However, in the construction step (Eq. 7), using the least-squared method causes large scale artifacts due to the many attenuation points distributed over scene. These artifacts are difficult to detect (see Supplementary Material). However, when we fix the attenuation points, the artifacts remain localized around the attenuation points and they are easily detected (Fig. 9).

Bas relief ambiguity. Viewing a 3D Lambertian surface orthographically creates ambiguity in determining its structure. A same appearance can be produced by different surfaces under appropriate lighting and with a correct viewpoint. This is called "bas relief ambiguity" [BKY99]. This ambiguity is not predominant on high reliefs since high reliefs keep the presence of separate scene elements at different depth levels (e. g., the arm in Fig. 12b). In contrast to bas reliefs, high reliefs can be viewed from a wider range of viewing angles and still preserve the appearance of the original shape. Fig. 12 shows a comparison of our high relief with a bas relief generated with the method of Weyrich et al. [WDB*07]. The appearance changes dramatically once the viewpoint of the bas relief is changed, whereas it is mostly maintained for our high relief.

Performance. Our method is not optimized for performance. For scenes having approximately one quarter million vertices and half a million faces, it takes 54 seconds to generate a high relief with $k = 128$. The execution time is dominated by the restoration operation which takes 34 seconds. The execution times were measured on a computer with an Intel Xeon E3-1225 processor.

Limitations. Our algorithm is limited to connected meshes. In some cases, a number of visible parts might stay behind the relief plane when the number k of attenuation points is small. Increasing k increases the constraints to keep all visible parts of the scene in front of the relief plane.

For complex scenes, having the parts of scenes which are not directly connected to each other may cause the interpenetration of these parts in the case of too high constraints induced by a large number of attenuation points.

5. Conclusion

We propose a method to create visually pleasing high reliefs from 3D scenes that is simple to operate while still allowing for creative control. Our method relies on the differential coordinates in order to preserve the overall shape, the depth discontinuities, and the fine details of the scene. We bring the overall shape towards the relief plane by using a number of attenuation points. First, new positions of the attenuation points are calculated, then new positions of all other vertices are found by using their differential coordinates. We give the freedom to artists to control the parameters adjusting the depth range, the attachment to the relief plane, and the range compression. The proposed high relief synthesizing method may find a widespread usage for decorative and home purposes as well as for architecture. With today's 3D printing facilities, the fabrication of high reliefs has been considerably simplified.

Our high relief method is a first step towards solving the problem of automatic generation of high reliefs and it can inspire several future works. The method may be extended to create high reliefs on arbitrary relief surfaces. One may create optical illusions by considering light changes, shadow effects, and forced perspective. In addition, full-color high relief design may become fashionable for house decorations and gifts.

Acknowledgements

The authors thank the anonymous reviewers for their useful comments and suggestions. The 3D models are taken from the following resources: Dragon, Asian Dragon, Happy Buddha, and Armadillo - The Stanford 3D Scanning Repository; Angel (Fig. 1) - http://www.cadnav.com; Status (Fig. Fig. 13-e) and Sculpture (Fig. Fig. 13-a) - http://www.archive3d.net.

References

[Ale03] ALEXA M.: Differential coordinates for local mesh morphing and deformation. *The Visual Computer 19*, 2 (2003), 105–114. 2, 3

[Arn74] ARNHEIM R.: *Art and visual perception: A psychology of the creative eye*. Univ of California Press, 1974. 3

[BH11] BIAN Z., HU S.-M.: Preserving detailed features in digital bas-relief making. *Computer Aided Geometric Design 28*, 4 (2011), 245–256. 2, 5

[BKY99] BELHUMEUR P. N., KRIEGMAN D. J., YUILLE A. L.: The bas-relief ambiguity. *International Journal of Computer Vision 35*, 1 (1999), 33–44. 8

[CMS97] CIGNONI P., MONTANI C., SCOPIGNO R.: Computer-assisted generation of bas-and high-reliefs. *Journal of graphics tools 2*, 3 (1997), 15–28. 2, 7

[DH04] D'AMELIO J., HOHAUSER S.: *Perspective drawing handbook*. Courier Dover Publications, 2004. 3

[FLW02] FATTAL R., LISCHINSKI D., WERMAN M.: Gradient domain high dynamic range compression. In *ACM Transactions on Graphics (TOG)* (2002), vol. 21, ACM, pp. 249–256. 2

[Hof39] HOFFMAN M.: *Sculpture inside and out*. Bonanza Books, New York, 1939. 2

[KG00] KARNI Z., GOTSMAN C.: Spectral compression of mesh geometry. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (2000), ACM Press/Addison-Wesley Publishing Co., pp. 279–286. 2, 3

[KTB*09] KERBER J., TEVS A., BELYAEV A., ZAYER R., SEIDEL H.-P.: Feature sensitive bas relief generation. In *Shape Modeling and Applications* (2009), IEEE, pp. 148–154. 2

[KWC*12] KERBER J., WANG M., CHANG J., ZHANG J. J., BELYAEV A., SEIDEL H.-P.: Computer assisted relief generation: A survey. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 2363–2377. 2, 5

[M*67] MACQUEEN J., ET AL.: Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (1967), vol. 1, California, USA, p. 14. 4

[MM72] MURRAY P., MURRAY L.: *A dictionary of art and artists*. Penguin Books, 1972. 1

[SBS07] SONG W., BELYAEV A., SEIDEL H.-P.: Automatic generation of bas-reliefs from 3d shapes. In *Shape Modeling and Applications* (2007), IEEE, pp. 211–214. 2

[SCOL*04] SORKINE O., COHEN-OR D., LIPMAN Y., ALEXA M., RÖSSL C., SEIDEL H.-P.: Laplacian surface editing. In *Proceedings of the 2004 Eurographics, ACM SIGGRAPH symposium on Geometry processing* (2004), pp. 175–184. 3, 5, 6, 8, 11

[Sor05] SORKINE O.: Laplacian mesh processing, http://www.cs.berkeley.edu/~jrs/meshpapers/Sorkine.pdf. *Eurographics STAR - State of The Art Report* (2005), 53–70. 5, 11

[SPSH14] SCHÜLLER C., PANOZZO D., SORKINE-HORNUNG O.: Appearance-mimicking surfaces. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH ASIA) 33*, 6 (2014), 216:1–216:10. 2

[SRML09] SUN X., ROSIN P. L., MARTIN R. R., LANGBEIN F. C.: Bas-relief generation using adaptive histogram equalization. *Visualization and Computer Graphics, IEEE Transactions on 15*, 4 (2009), 642–653. 2

[WDB*07] WEYRICH T., DENG J., BARNES C., RUSINKIEWICZ S., FINKELSTEIN A.: Digital bas-relief from 3d scenes. In *ACM Transactions on Graphics (TOG)* (2007), vol. 26, ACM, pp. 32–32. 2, 4, 8

APPENDIX

I. Notation

$\mathbf{M}_{e\times f}$ denotes a matrix $\mathbf{M}$ where $e \times f$ is the dimension of the matrix. $\mathbf{M}(a:b, c:d)$ denotes a sub-matrix of the matrix $\mathbf{M}$ where $a:b$ denotes rows a to b and $c:d$ denotes columns from c to d.

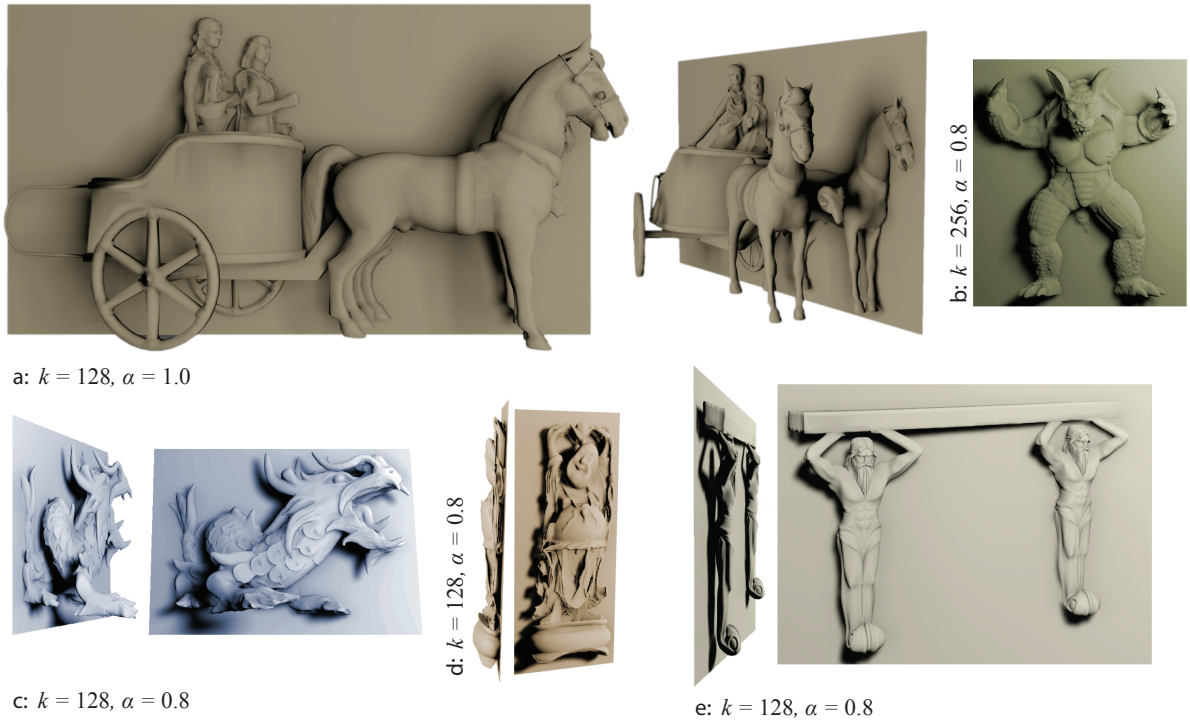
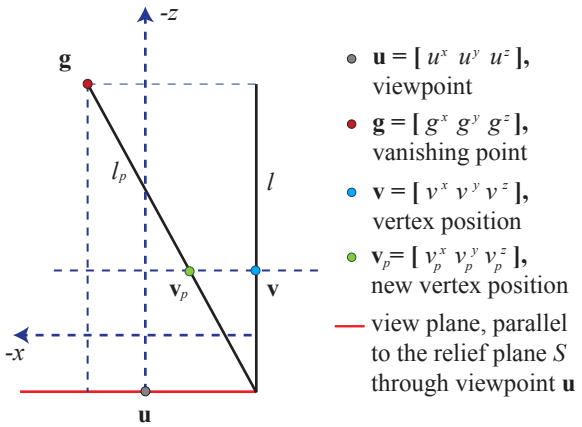


Figure 13: *High relief examples for different type of scenes.*

II. One point perspective deformation for forced perspective

The perspective deformation consists in bringing line l into line l_p obtained by connecting the projection of l on the view plane with vanishing point **g**.



A vertex **v** is displaced to $\mathbf{v}_p$ according to the following relationship:

$$\frac{v^x - v_p^x}{v^x - g^x} = \frac{v^z - u^z}{g^z - u^z}$$

Solving for v_p^x yields

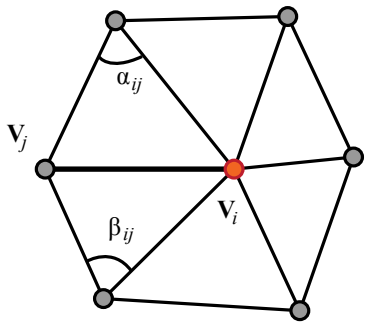
$$v_p^x = v^x - \frac{(v^x - g^x) \cdot (v^z - u^z)}{g^z - u^z}.$$

Since similar considerations apply to the y and z; we obtain

$$v_p^y = v^y - \frac{(v^y - g^y) \cdot (v^z - u^z)}{g^z - u^z}$$

$$v_p^z = v^z.$$

III. Cotangent weights



For a vertex $\mathbf{v}_i$, the differential coordinates $\Delta\mathbf{v}_i$ are defined as follows;

$$\Delta\mathbf{v}_i = \frac{1}{|\,\psi\,|}\sum_{j\in N}\omega_j(\mathbf{v}_i-\mathbf{v}_j)$$
$$\omega_j = \frac{1}{2}(\cot\alpha_{ij}+\cot\beta_{ij})$$

where N denotes the set of neighbour vertices (i.e. the 1-ring) on the mesh surface around $\mathbf{v}_i$. ω_j is the cotangent weight for neighbour vertex $\mathbf{v}_j$ and $|\,\psi\,|$ is the Voronoi area acting as normalization factor [Sor05].

IV. New relief distances of attenuation points

In order to find differential values and minimize Eq 5, we proceed as recommended by Sorkine et al [SCOL*04] and use matrix form $\Delta\mathbf{h} = \mathbf{L}\mathbf{h}$, where $\mathbf{h}$ and $\Delta\mathbf{h}$ are vectors of length k representing the relief distances a_i^z of all attenuation points and their differential relief distances Δa_i^z respectively. Matrix $\mathbf{L}$, computed as $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$ and called the Laplacian matrix, is a $k \times k$ sparse matrix. $\mathbf{A}$ is the adjacency matrix giving the neighboring attenuation points, where neighbours of an attenuation point are represented with the value 1 and non-adjacent ones with the value 0. Matrix $\mathbf{D}$ defined as $\mathbf{D} = diag\{d_1,...,d_n\}$ is the degree matrix with d_i denoting the degree of vertex $\mathbf{a}_i$, i.e. the number of neighbour attenuation points. After calculating the reduced differential relief distances $\Delta\mathbf{h}'$ according to Eq. 3, we calculate the new relief distances $\mathbf{h}'$ of attenuation points by solving equation $\Delta\mathbf{h}' = \mathbf{L}\mathbf{h}'$. The new relief distances $\mathbf{h}'$ are recovered by introducing our constraints into the system. With the constraints indexed from b to k as indicated in Eq. 4, the system to solve by the least square method is the following:

$$\left(\begin{array}{c|c} \mathbf{L}_{k\times k} \\ \hline 0 & \mathbf{I}_{(k-b+1)\times(k-b+1)} \end{array}\right)\mathbf{h}' = \begin{pmatrix}(\Delta\mathbf{h})' \\ 0\end{pmatrix}$$

V. Reconstruction of scene vertices

We use matrix form $\Delta\mathbf{V} = \mathbf{L}\mathbf{V}$, where $\mathbf{V}$ and $\Delta\mathbf{V}$ are $n\times 3$ matrices representing the vertices V and their differential coordinates ΔV. Laplacian matrix $\mathbf{L}$, a $n\times n$ matrix, is computed as $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$. $\mathbf{A}$ is the adjacency matrix giving the vertex connectivity, where vertices adjacent to one vertex are represented with their cotangent weight and non-adjacent ones with the value 0. Matrix $\mathbf{D} = diag\{d_1,...,d_n\}$ is the degree matrix with d_i denoting the Voronoi area of vertex $\mathbf{v}_i$. We insert our constraints (i. e., attenuation points $\mathbf{V}'(m:n)$) described in Eq. 6 into the system and solve it for all other vertices [Sor05].

$$\left(\begin{array}{c|c} \mathbf{L}(1:m-1,1:n) \\ \hline 0 & \mathbf{I}_{(n-m+1)\times(n-m+1)} \end{array}\right)\mathbf{V}'(1:n,3) = \begin{pmatrix}\Delta\mathbf{V}(1:m-1,3) \\ \mathbf{V}'(m:n,3)\end{pmatrix}$$

We only solve for the z coordinates $\mathbf{V}'(1:n,3)$, since x and y do not change.

VI. Restoration

We apply a blending between higher order differential coordinates $\Delta^2\mathbf{V}$ and first order differential coordinates $\Delta\mathbf{V}$ to obtain blended differential coordinates $\Delta\mathbf{V}'$. By using the Laplacian matrix described in Appendix V: $\Delta\mathbf{V}' = t(\mathbf{L}\mathbf{V}) + (1-t)(\mathbf{L}(\mathbf{L}\mathbf{V})) = (t\mathbf{L} + (1-t)\mathbf{L}\mathbf{L})\mathbf{V} = \mathbf{L}'\mathbf{V}$, where t is the blending variable ($t = 0.9$). Let us index the new positions of vertices $\mathbf{V}'$ with the first r vertices having high errors (Eq. 8). Then we recompute the values of these r vertices by adding the ones not having errors as positional constraints in the system:

$$\left(\begin{array}{c|c} \mathbf{L}'(1:r,1:n) \\ \hline 0 & \mathbf{I}_{(n-r+1)\times(n-r+1)} \end{array}\right)\mathbf{V}''(1:n,3) = \begin{pmatrix}\Delta\mathbf{V}'(1:r,3) \\ \mathbf{V}'(r+1:n,3)\end{pmatrix}$$

where $\mathbf{V}''$ yields the final positions of the scene vertices. We only solve for the z coordinates $\mathbf{V}''(1:n,3)$ since x and y do not change.