Sculpture Paintings

Sami Arpa
School of Computer and Communication Sciences (IC)
Ecole Polytechnique Federale de Lausanne (EPFL)
sami.arpa@epfl.ch

Sabine Süsstrunk
School of Computer and Communication Sciences (IC)
Ecole Polytechnique Federale de Lausanne (EPFL)
sabine.susstrunk@epfl.ch

Roger D. Hersch
School of Computer and Communication Sciences (IC)
Ecole Polytechnique Federale de Lausanne (EPFL)
rd.hersch@epfl.ch

Figure 1: A sculpture painting produced by our framework. The scene is partly in 2D and partly in 3D. Up to a certain viewing angle, 2D and 3D scene elements are perceived as being part of the same composition.

ABSTRACT

We present a framework for automatically creating a type of artwork in which 2D and 3D contents are mixed within the same composition. These artworks create plausible effects for the viewers by showing a different relationship between 2D and 3D at each viewing angle. As the viewing angle is changed, we can clearly see 3D elements emerging from the scene. When creating such artwork, we face several challenges. The main challenge is to ensure the continuity between the 2D and the 3D parts in terms of geometry and colors. We provide a 3D synthetic environment in which the user selects the region of interest (ROI) from a given scene object to be shown in 3D. Then we create a flat rendering grid that matches the topology of the ROI and attach the ROI to the rendering grid. Next we create textures for the flat part and the ROI. To enhance the continuity between the 2D and the 3D parts of the object, we include bas-relief profiles around the ROI. Our framework can be used as a tool in order to assist artists in designing such sculpture paintings. Furthermore, it can be applied by amateur users to create decorative objects for exhibitions, souvenirs, and homes.

Expressive '18, August 17–19, 2018, Victoria, BC, Canada

ACM ISBN 978-1-4503-5892-7/18/08…$15.00
https://doi.org/10.1145/3229147.3229156

CCS CONCEPTS

• **Computing methodologies** → **Mesh geometry models**; • **Applied computing** → **Fine arts**;

KEYWORDS

Sculpture paintings, bas reliefs, high reliefs, color reliefs, 3D printing

ACM Reference Format:
Sami Arpa, Sabine Süsstrunk, and Roger D. Hersch. 2018. Sculpture Paintings. In *Expressive '18: The Joint Symposium on Computational Aesthetics and Sketch Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering, August 17–19, 2018, Victoria, BC, Canada.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3229147.3229156

1 INTRODUCTION

In this paper, we define a framework for blending a 2D projection and 3D content within a single artistic composition that we call sculpture paintings. Such artistic forms are created by artists as a combination of paintings and colorful sculptures. Some scene elements are flat, like paintings, while others emerge from the 2D plane as 3D objects, as shown in Figure 2. This creates interesting visual effects for the observers.

Artists who blend a painting and a sculpture must meet several challenges. First, the geometric visual information of a painting does not fundamentally change as a function of the viewing position while a sculpture reveals new information at each viewing angle. In the sculpture paintings, each new viewing position creates a new composition of 2D and 3D scene elements. Creating a composition that provides plausible percepts for many different viewing angles is challenging. In addition, the textures of the 2D and 3D parts need to be blended. Lighting can dramatically change the perception of

Figure 2: A sculpture painting created by the artist Marshall Mithourad.

the 3D elements. Depending on the type of light and its position, the diffuse shading, specular highlights, and shadows can change to a great extent the appearance of the 3D elements. Contradictions with the 2D part may arise.

The challenges encountered by artists also constitute the challenges of the present work. Similar to artists, we try to create a seamless continuity between the 2D and the 3D parts of the surface, allowing the user to perceive plausible emerging 3D elements from a variety of viewing positions.

The initial challenge consists of matching the topologies and tessellations of the 3D part and the 2D part to create a continuous attachment. One solution is to regenerate the 3D part as an extension of the plane i. e., as a bas-relief. However, bas-reliefs cannot correctly represent 3D shapes, since they rely on elevations from the surface. Instead, we cut the region of interest directly from the scene, create a grid that has homeomorphic boundaries and a similar topology as the 3D part, and attach it to the grid. A simple cut operation by intersecting the 3D mesh with a plane creates a boundary polygon that provides a smooth transition only if the result is observed in the camera direction. However, this yields jagged edges once the final result is perceived along a direction different from the camera direction. Therefore, we generate a cut polygon which looks smooth on both camera direction and orthogonal directions.

The second challenge is to match shading and textures. The 2D and 3D elements of an object have to match perfectly to look as parts of the same composition. The sudden change in the surface normals between the flat part and the 3D part creates a big conflict apparent at the boundaries. In order to create smoother transitions, we introduce bas relief profiles on the flat part of the object. Bas-relief is a type of relief (sculpture) where all forms are represented as elevations from a plane. Bas relief profiles not only provide a smooth transition around boundaries, they also ensure consistency between the appearance properties of the flat part and 3D part especially in case of specularities. Furthermore, we define pre-lighting conditions for baking the textures of the scenes and we perform style transfer [Reinhard et al. 2001] between the scene background texture and the object texture.

We initially take a 3D scene as our input and render it onto a plane that we call "rendering plane" (Fig. 3). Then we allow the user to select on an object a region of interest (ROI), delimited by a "cut plane". By refining the mesh along the boundary of the cut position, we extract the region of interest (ROI) in a seamless way. We then locate the boundary vertices on the rendering plane and create a grid on the rendering plane that consists of the projection of the boundary of the ROI. After attaching the ROI and the rendering grid along the boundary vertices, we create bas relief profiles on the non-ROI flat part in order to ensure a smooth transition between the flat part and the 3D part. Finally, we create the textures and map them onto the flat part and the 3D part. The final result (Fig. 1) is a surface that is formed by a fully connected mesh.

Our results can be printed in 3D thanks to the recent developments in multi-color 3D printing. The printed results can be used for decorations and exhibitions.

2 RELATED WORK AND BACKGROUND

Automatic generation of artwork by computer algorithms has been a significant research topic in a variety of domains. The goal of some of these works is to create tools for the artists while others generate artworks in a completely automated manner. In computer graphics, the focus has been mostly on painterly rendering [Hertzmann 2010] and relief generation [Kerber et al. 2012]. Our work is related to the domain of relief generation.

Relief is a sculptural technique where the sculpted elements are attached to a background surface. There are different types of reliefs depending on the attachment type of the sculpted part. If the sculpted parts are designed only as the elevations from the surface without undercuts, the relief is called bas-relief (low-relief). If parts detach from the surface as 3D forms, the relief is called high-relief.

In computer graphics, the main approach for designing a bas relief is based on a height-field [Cignoni et al. 1997]. For a given camera position, the input scene is captured as a height-field. This then defines the elevation ranges on a regular mesh grid. The main challenge of the bas relief methods consists in mapping the height-field of the scene to a limited range while preserving the details of the original scene. Initial methods [Cignoni et al. 1997] use non-linear compression operations. However, they cannot generate plausible reliefs for higher depth range variations. Recent methods [Kerber et al. 2009; Song et al. 2007] use the gradient domain [Fattal et al. 2002] to compress the height field. These methods preserve most details of the scene for very limited height ranges. Weyrich et al. [2007] use a multi-scale gradient compression, which results in very plausible bas-reliefs.

For the high relief, the height field representation of the bas relief is not applicable. Instead, the 3D scene geometry should be directly connected with the relief plane. Schüller et al. [2014] introduce a generalization of bas relief synthesis for arbitrary target surfaces. Since this method does not require a height field as input data but uses the 3D scene geometry, it can also be used to generate high reliefs. A recent work by Arpa et al. [2015] creates high reliefs that can be viewed from a wide range of viewing angles with an appearance similar to the original 3D shape. Well chosen control points enable attaching the scene to the relief plane.

Our work has similarities to high reliefs. However, the challenges are different. Although, in both, the 3D parts are attached to a plane, in high reliefs there is no transition between the planar projected elements and the 3D elements. All forms are in 3D within a limited space. In the present work, since some parts of the scene are projected onto the rendering plane, the method of attachment should allow a seamless continuity between 2D and 3D regions, which is achieved with the addition of bas relief profiles.

Our method is also related to 3D mesh transplanting. Sorkine et al. [2004] describe how to transplant one part of a 3D mesh to another 3D mesh. Their method includes two main processes. First, two meshes are topologically refined. This includes creating a consistent triangulation of the two meshes in order to match the boundaries of the parts that will be transplanted. Then, differential coordinates are used to create a smooth transition of the geometry, after attachment of the two meshes.

3 APPROACH

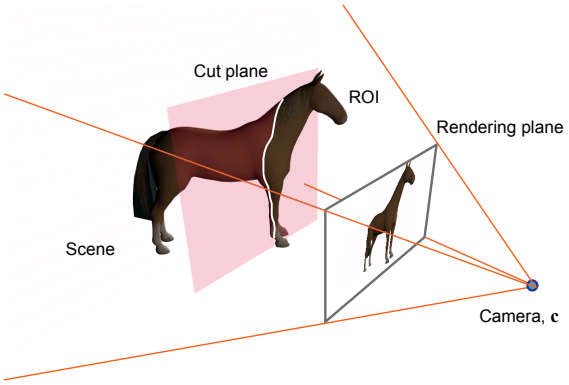

Figure 3: The overall setup of the scene. We define a camera position and a rendering plane. For a given object in the scene, the user selects, through a cut plane, the object part (ROI) to be in 3D.

We propose a synthesis technique where parts of a given 3D scene are rendered on a plane and the user selected parts emerge as 3D forms from the plane. The final geometry is a fully connected mesh where 2D parts and 3D parts are within the same continuum. In order to provide a better intuition, we define two separate planes: a rendering plane that represents the final design and a cut plane which provides a user control over the selection of the region of interest.

Let us introduce the notations used hereinafter. We consider the rendering plane $P \in R^3$ and a 3D scene $M = \{V, F, E\}$, 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 $\mathbf{v}_i = [v^x\ v^y\ v^z]$. F denotes the set of faces, where each face $\mathbf{f}_i$ is a triangle. E denotes the set of edges that form the faces. The scene is further defined by a camera position $\mathbf{c}$. We cut the scene M with the user controlled cut plane H to determine the region of interest (ROI), which is denoted by $R = \{V_r, F_r, E_r\}$. A sample setup of the scene is given in Fig. 3.

After the user selects the position of the cut plane, we first locate the ROI $R = \{V_r, F_r\}$ by finding the corresponding faces from the original scene object M. Then we refine the boundary of the ROI in order to create a smooth cut line. After creating the rendering grid, we attach the ROI to the rendering plane. Then we add bas relief profiles to the flat part of the object to create a smooth transition between the 2D and the 3D elements. Finally we generate the textures and UV maps to obtain the final result. In the following, we detail each step.

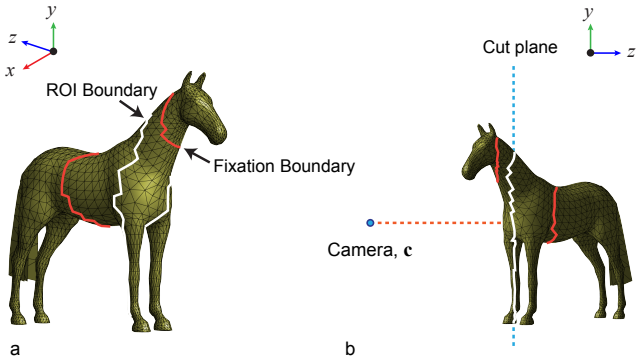

Figure 4: The ROI boundary and fixation boundaries for the sample scene.

3.1 Extraction of ROI and boundary

The ROI is the 3D part of a scene object that is to be represented both by 2D parts and 3D parts. In order to extract the ROI, we first find a boundary on the input scene object aligning with the cut plane, refine this boundary to create a smooth polygon, and finally detach the boundary. A simple solution such as simply intersecting the 3D mesh with a plane yields jagged edges once the final result is viewed from directions other than the camera direction. Our method generates a boundary polygon ensuring the perception of a smooth transition both when viewed along the camera direction and other directions.

We define all faces of the scene that lay between the rendering plane and the cut plane as the ROI. The cut plane is parallel to the rendering plane and its position is given by the user as a distance h_z from the camera. Since our camera lies on the z axis of the world coordinate system, the z position of the cut plane h_z is considered as the parameter governing the transition between the 2D projected part of the object and the 3D part forming the ROI. Each face $\mathbf{f}_i$ is defined as a ROI face if all of its vertices satisfy the following condition:

$$v_i(z) \leqslant h_z + \beta \cdot \eta \tag{1}$$

where $\beta \cdot \eta$ is the tolerance parameter, η is the average edge length in the scene and β is the control parameter, $\beta = 0.2$. The tolerance parameter enables including faces into the ROI that have vertices close to the cut plane.

Next, we find the boundary vertices $B = (v_1^b, ..., v_m^b)$ of the ROI, where $B \subset V$ and m is the number of boundary vertices. Boundary

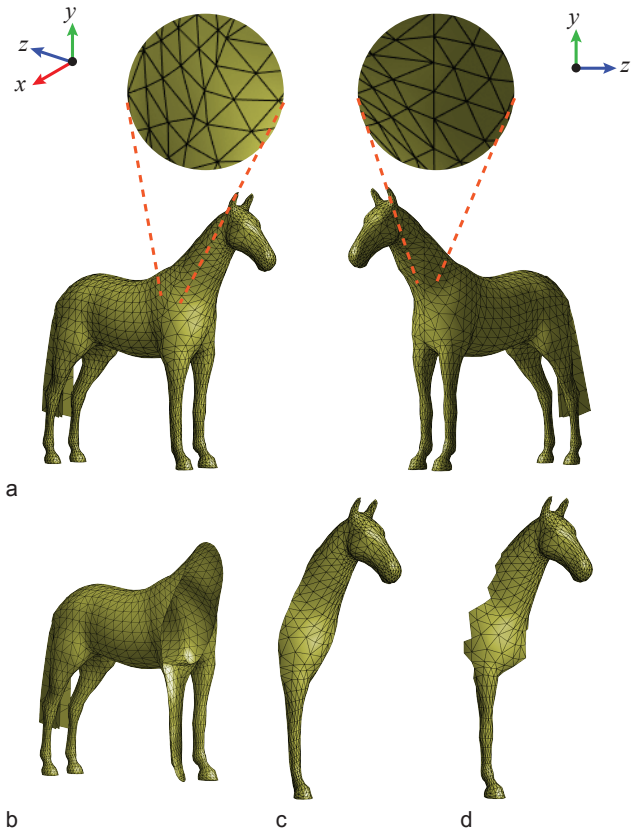

Figure 5: a) The ROI boundary after the refinement operation. b) The non-ROI part after the scene object is cut. c) The ROI part after the scene object is cut along the boundary. d) The ROI without boundary refinement.

edges are referenced only by a single triangle in the mesh. In other words, a boundary edge cannot be a common edge for more than one triangle. In order to find the boundary edges $E' \subset E$, we scan through all triangles in the mesh and take the edges having a single reference count. By using the edge set E, we do this efficiently in $O(n)$ complexity. With a simple traversal of E', we then convert the edge set to an ordered polygon loop (Fig. 4).

As it can be seen in Fig. 4b, the ROI boundary is not perfectly aligned with the cut plane due to the structure of the original geometry. Leaving the ROI like that would result in continuity problems regarding the geometry and texture when attaching the ROI to the rendering plane. Therefore, we refine the ROI boundary to be aligned with the cut plane and smooth it to ensure that its projection on the rendering plane is a smooth polygon, showing a smooth transition when viewed from many viewing directions.

We use differential coordinates to make deformations on the ROI boundary as rigid as possible. We calculate the differential coordinates Δv_i for each vertex v_i as described by Sorkine et al. [Sorkine 2005].

Then, we change the boundary positions v^b. We first change the z coordinates of all the ROI boundary vertices to the z coordinate h_z of the cut plane. Next we smooth the x and y coordinates of the ROI boundary vertices separately by applying a 1D Gaussian kernel on the boundary polygon B.

To limit the deformations of the scene object, we define two additional boundaries that are called "fixation boundaries" as shown in Fig. 4. We put their original positions v^f as constraints into our optimization function. As such, only the vertices between the fixation boundaries change.

Then we solve the following optimization function to obtain the refined geometry of the 3D scene:

$$E(v_1, ..., v_n) = \sum_{i=1}^{n} \|\Delta v_i' - \Delta v_i\|^2 + \sum_{j=1}^{m} \|v_j^b{}' - v_j^b\|^2 + \sum_{k=1}^{d} \|v_k^f{}' - v_k^f\|^2 \quad (2)$$

The first term in the optimization minimizes the square differences between original differential coordinates Δv_i and the differential values of the refined geometry $\Delta v_i'$. The second term minimizes the differences between the smoothed m boundary positions v^b and the final boundary positions on the refined geometry. The third term minimizes the variations on the d fixation boundaries v^f. This optimization problem is solved as a sparse linear systems of equations. We thus obtain the refined scene with the smoothed boundary, as illustrated in Fig. 5.

3.2 Preparation of the rendering plane

We create a grid to represent the rendering plane and define attachment points for the ROI. First, we find the projection of the ROI boundary on the rendering plane. Then we find the average boundary edge length a on the rendering plane.

$$a = \frac{\sigma}{m} \sum_{i=1}^{m} \|b_i' - b_{i+1}'\| \qquad q = w/a \qquad r = h/a \quad (3)$$

where σ is the density factor, q is the number of columns, r is the number of rows, w and h are the width and the height of the rendering plane, respectively, and b' denotes a projected vertex of the ROI boundary.

We create $q \times r$ points that are equally distributed over the rendering plane as shown in Fig. 6a. Then, we add the projected boundary vertices on the grid and remove all vertices inside the polygon that is formed by the projected boundary (Fig. 6b). We also eliminate all points that have a distance to the boundary polygon of less than $\frac{a}{2}$.

Then, we apply a Delaunay triangulation [Su and Drysdale 1997] over the grid points to obtain a mesh surface (Fig. 6c). Finally we remove all triangles that reside within the boundary polygon. For this purpose, we remove all faces whose vertices are all boundary vertices. After this operation, we obtain our final rendering plane which is ready for attachment with the ROI (Fig. 6d).

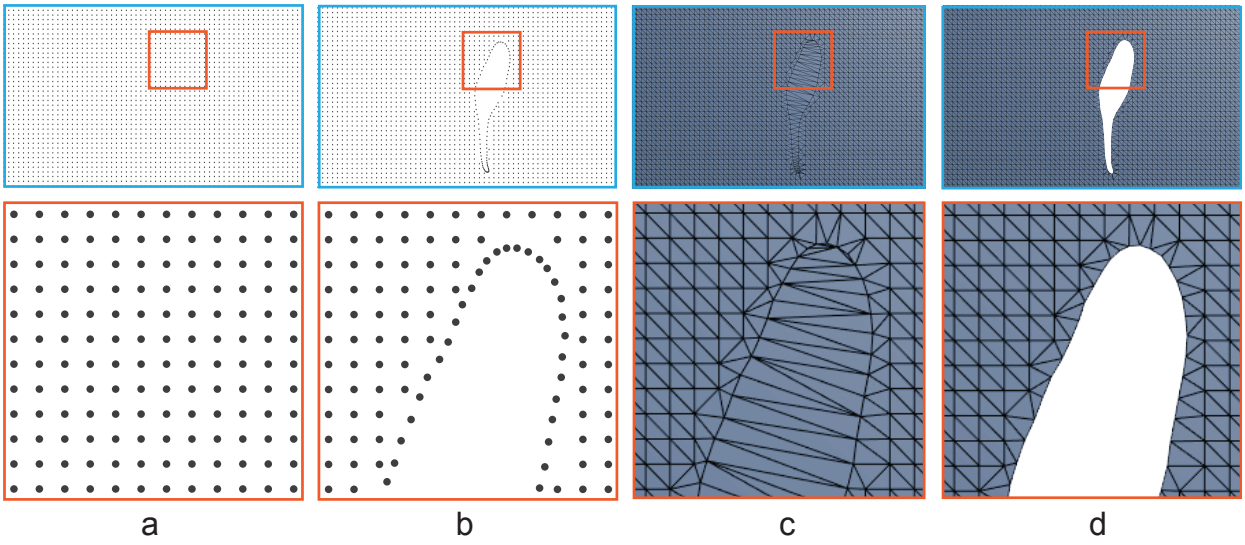

Figure 6: a) The rendering grid points. b) Boundary vertices are added and the vertices inside the boundary polygon are removed. c) The initial geometry is created. d) Triangles inside the boundary polygon are removed.

3.3 Attaching the ROI

In order to get a final fully connected surface, we need to attach the ROI to the rendering grid . First we transform the ROI in order to match the boundary of the ROI and the boundary that is projected on the rendering grid. This transformation involves scaling the ROI and translating it to the corresponding position.

After aligning he boundary vertices on the rendering grid and the ROI boundary on the same positions, we merge the vertices and faces of the ROI with the vertices and faces of the rendering grid. Finally, we unite all boundary vertices and create the same index references to them from the face list. In this way, we obtain a final surface in which the ROI is fully connected to the rendering plane along its boundary (Fig. 7).

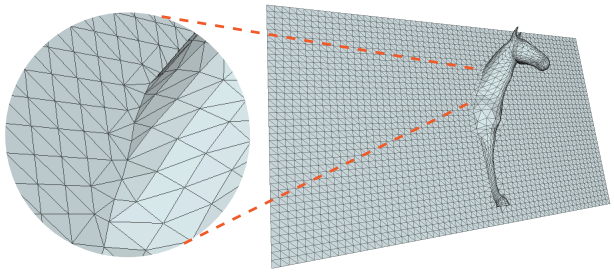

Figure 7: Final geometry after the attachment operation.

3.4 Incorporating Bas Relief Profiles

In order to create a smoother transition between the 2D and the 3D parts of the scene object, we incorporate bas relief details onto the 2D region and refine the attached ROI accordingly.

First, we find the corresponding bas relief elevations on the rendering grid. To do that, we extract the depth map of the non-ROI scene object and find the corresponding elevations for each grid point v_i^d. Then, similar to the method of Weyrich et al. [2007], we calculate the gradients $(\frac{\partial v_i^d}{\partial x}, \frac{\partial v_i^d}{\partial y})$ of each grid point by taking the backward differences with its neighbouring vertices in x and y directions. Then we use the magnitude of the gradients to find silhouette vertices:

$$\|(\frac{\partial v_i^d}{\partial x}, \frac{\partial v_i^d}{\partial y})\| > s \tag{4}$$

where any vertex that has a larger gradient magnitude than the silhouette threshold s is considered as a silhouette vertex.

Next we assign depth values v_i^d to the flat part of the attached geometry as z coordinates. We find differential coordinates of the new attached geometry as discussed in Section 3.1. Then, we change the z coordinate of silhouette vertices v_i^s to ensure that they have zero elevation from the rendering plane and set them as constraints in our optimization function.

$$\begin{aligned} E(v_1(z), ..., v_n(z)) = &\sum_{i=1}^{n} \|\Delta v_i'(z) - \Delta v_i(z)\|^2 \\ &+ \sum_{j=1}^{h} \|v_j^s\prime(z) - v_j^s(z)\|^2 \end{aligned} \tag{5}$$

where h is the number of silhouette vertices. The first term of the optimization ensures preserving the differential values of the original ROI and of the newly assigned depth elevations on the flat part. The second term ensures that the silhouette vertices do not have any elevation from the surface. Then we solve the sparse linear system to obtain the final result (Fig. 8). In contrast to previous

approaches [Weyrich et al. 2007], we do not need to apply scaling on the gradients for a smooth transition between the 2D and the 3D parts.

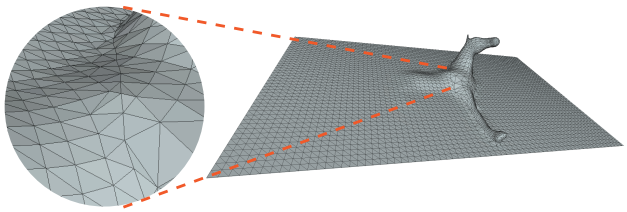

Figure 8: Addition of bas relief profiles.

3.5 Textures

We use two different texture maps for the final geometry. One is defined for the 2D part of the rendering grid including the bas relief and the second is for the 3D part, the ROI.

For the 2D part of the rendering grid, we use our main scene (Fig. 3). After replacing the scene with the non-ROI (Fig. 5b), we render it on the rendering plane. We use the rendered 3D scene image as our texture map for the flat part. UV coordinates are calculated by simply scaling the original x and y coordinates of the rendering plane in the interval between 0 and 1.

For the 3D part, we use the original texture map and UV coordinates. Since we keep the original geometry and connectivity for the 3D part, original UV coordinates do not create any problem except negligible deformations along the boundary.

To avoid the lighting contradictions between the ROI and the flat region, we bake the shading effects present in the scene onto the original texture of the scene object [Summers 2004, Chapter 7]. The texture baking enables embedding lighting effects of a 3D geometry into a 2D image texture. First, we set the lighting conditions and bake the texture of the scene object. Then, for rendering the non-ROI part, we only use ambient light with the newly baked texture. For the 3D part, the baked texture is directly used.

4 RESULTS AND DISCUSSION

As shown in Fig. 9, we can generate a great variety of sculpture paintings. The results can be obtained with minimal user intervention as soon as an appropriate scene and cut plane position is provided. A great effort would be required from an artist to render the 2D part and match the 3D part over the same plane. We automatically create a smooth transition both in respect to the geometry and to the texture.

On high tessellation meshes, our method works better since the boundary refinement causes less deformations. We also create a planar grid having a tessellation similar to the ROI. Scene objects having very low tessellations yield planar grids with very low tessellations that cannot represent bas relief details.

4.1 Lighting artefacts

The geometry differences between flat and 3D parts can create lighting artefacts. Lighting artefacts arise for several reasons. First, they can be caused by abrupt changes of the normal directions in the transition between the 3D part and the flat part. Second, the difference of the tessellations between the 2D and 3D parts can reveal discontinuities. Third, for glossy materials, specular variations from the 3D part would not occur on the flat part. One of the biggest challenge of this work was to minimize such artefacts. Introducing bas relief profiles solves the first and the third problems to a great extent. By having bas relief profiles, we have a smoother transition from 3D to 2D and also create small elevations on the flat part which result in similar specular variations on the flat and the 3D parts. For the second problem, we generate a rendering grid having tessellations similar to the 3D part as described in Section 3.2. Furthermore, we bake the textures of the scene with pre-defined lighting conditions. In this way, we ensure that the rendered 2D part and 3D part have matching shadings. These operations decrease to a large extent lighting artefacts (Fig. 9c). The produced sculpture paintings look good under many lighting conditions and the eye can be fooled from many viewing directions.

4.2 Viewing angle ambiguity

Our results can be observed from a great variety of angles. Each angle provides a different interpretation of the composition. However, the limit of view independence depends on the content. The scenes including elastic and self-moveable content such as a horse, a deer, or a shark have larger view-independence than the scenes with solid elements such as cars, buildings, and furniture. When the viewer changes its position, the new composition is perceived as if the object has changed its pose. However, since this kind of self movement is not possible for solid objects, they might be perceived as broken or deformed (Fig. 9f, bottom). Sculpture paintings are not as ambiguous as bas reliefs. The 3D part of the object is the exact replica of the original geometry, which is perceptually dominant within the composition. Therefore, the object's shape can be perceived from many angles.

4.3 Scene design and cut position

Our camera position and rendering plane is predefined. Users may modify the input scene by changing its position, scaling factor, and rotation angle. We also allow the user to control the cut plane position in order to specify the desired ROI forming the 3D part of the scene object. In most of the objects, it is possible to select many different positions (Fig. 9a-b). One important criteria in the cutting position is to have a topology that allows a homeomorphic relationship with the rendering plane. In practice, this means having a single boundary polygon on the cut position. However, our method can be generalized to support the attachment for multiple boundary polygons. The position of the cut plane provides an artistic control since each cut position creates a slightly different composition.

4.4 Style transfer

In addition to the main scene object, other objects in the scene help creating a supporting environment. We provide an interactive environment allowing designers to control their scene. Because of performance reasons, instead of creating a whole environment, background images may be used as well. In order to match the style of the background image and the textures of the scene, we

Figure 9: a-b) Sculpture paintings generated with the horse model according to different cut positions. c) Strong directional light can create artefacts at the boundary of 2D and 3D scene elements (bottom), but the addition of bas relief profiles decreases these artefacts to a large extent (top). d) Sculpture painting generated with the shark model. e) Sculpture painting generated with the woman model. f) Sculpture painting generated with the car model. (Best viewed on screen.)

apply a style transfer between background images and baked scene textures. We use a simple statistical analysis to transfer background color characteristics onto the textures [Reinhard et al. 2001].

4.5 Manufacturing

We can manufacture sculpture paintings thanks to the recent developments in multi-color 3D printing. One of the limitations is the narrow gamut range, which yields more dull colors compared to

Figure 10: Sculpture paintings manufactured with a 3D printer.

inkjet printing on paper. Yet, we can generate nice results with the current technology (Fig. 10).

5 CONCLUSION

We propose a method to create visually convincing sculpture paintings from given 3D synthetic scenes. We allow the user to choose the parts to be in 2D and 3D. Then we create a new geometry and textures that enable a smooth transition between the 2D and the 3D parts of a scene object. Although, at a first glance, merging 3D and 2D seems straightforward, creating convincing sculpture paintings that can be viewed for many different viewing angles under different lighting conditions is very challenging. To create such a geometry, we first refine the input geometry to extract the ROI along a cut plane. Then we create a rendering grid having a geometry compatible with the ROI. We incorporate bas relief details on parts of the rendering grid after attaching the ROI to the rendering grid. Finally we add the textures for both the flat part including the bas-relief element and the 3D part. We give the freedom to artists to design the overall input scene and to select the region of interest. The proposed method may find widespread usage in architecture, decoration, and households. Thanks to recent technologies in multi-color 3D printing, the fabrication of sculpture painting has been considerably simplified. The combination of 2D and 3D elements within a single composition offers new possibilities to artists and designers. Objects such as souvenirs or decorative elements incorporating sculpture paintings can be easily conceived and produced.

ACKNOWLEDGMENTS

The authors thank the anonymous reviewers for their useful comments and suggestions. The original 3D scenes used in the paper are obtained from the TURBOSQUID with the royalty free licences. https://www.turbosquid.com/

REFERENCES

Sami Arpa, Sabine Süsstrunk, and Roger D. Hersch. 2015. High Reliefs from 3D Scenes. *Computer Graphics Forum (Proc. EUROGRAPHICS 2015)* 34, 2 (2015), 253–263.

Paolo Cignoni, Claudio Montani, and Roberto Scopigno. 1997. Computer-assisted generation of bas-and high-reliefs. *Journal of graphics tools* 2, 3 (1997), 15–28.

Raanan Fattal, Dani Lischinski, and Michael Werman. 2002. Gradient domain high dynamic range compression. In *ACM Transactions on Graphics (TOG)*, Vol. 21. ACM, 249–256.

Aaron Hertzmann. 2010. Non-photorealistic rendering and the science of art. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*. ACM, 147–157.

Jens Kerber, Art Tevs, Alexander Belyaev, Rhaleb Zayer, and H-P Seidel. 2009. Feature sensitive bas relief generation. In *Shape Modeling and Applications*. IEEE, 148–154.

Jens Kerber, Meili Wang, Jian Chang, Jian J Zhang, A Belyaev, and H-P Seidel. 2012. Computer Assisted Relief Generation: A Survey. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 2363–2377.

Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. 2001. Color transfer between images. *IEEE Computer graphics and applications* 21, 5 (2001), 34–41.

Christian Schüller, Daniele Panozzo, and Olga Sorkine-Hornung. 2014. Appearance-Mimicking Surfaces. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH ASIA)* 33, 6 (2014), 216:1–216:10.

Wenhao Song, Alexander Belyaev, and H-P Seidel. 2007. Automatic generation of bas-reliefs from 3d shapes. In *Shape Modeling and Applications*. IEEE, 211–214.

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

Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and H-P Seidel. 2004. Laplacian surface editing. In *Proceedings of the 2004 Eurographics, ACM SIGGRAPH symposium on Geometry processing*. 175–184.

Peter Su and Robert L Scot Drysdale. 1997. A comparison of sequential Delaunay triangulation algorithms. *Computational Geometry* 7, 5-6 (1997), 361–385.

D. Summers. 2004. *Texturing: Concepts and Techniques*. Charles River Media. https://books.google.com.tr/books?id=-yO032NLCdoC

Tim Weyrich, Jia Deng, Connelly Barnes, Szymon Rusinkiewicz, and Adam Finkelstein. 2007. Digital bas-relief from 3D scenes. In *ACM Transactions on Graphics (TOG)*, Vol. 26. ACM, 32–32.