

# **New Approaches to Artistic Synthesis Relying on Visual Perception**

**Thèse N° 9189**

Présentée le 12 avril 2019

à la Faculté informatique et communications  
Laboratoire d'images et représentation visuelle  
Programme doctoral en informatique et communications

pour l'obtention du grade de Docteur ès Sciences

par

**Mahmut Sami ARPA**

Acceptée sur proposition du jury

Prof. M. Pauly, président du jury  
Prof. S. Süssstrunk, Prof. R. Hersch, directeurs de thèse  
Prof. P. Cignoni, rapporteur  
Prof. K. Myszkowski, rapporteur  
Prof. W. Jakob, rapporteur

2019



Imagination is more important than knowledge. For knowledge is limited, whereas imagination embraces the entire world, stimulating progress, giving birth to evolution.

— Albert Einstein



# Acknowledgements

First and foremost I want to thank my advisors, Roger D. Hersch and Sabine Süsstrunk. I appreciate all of their efforts and contributions that made my PhD experience very productive and creative. The freedom that they provided me for interdisciplinary works was very encouraging and it led me to discover new avenues in the field of computer graphics. Their guidance and support were extremely valuable; they helped me improve the quality of my work and my research skills.

I thank professors Mark Pauly, Wenzel Jakob, Karol Myszkowski, and Paolo Cignoni for accepting to become members of my PhD jury. I appreciate their helpful comments and suggestions on my thesis.

I appreciate the critiques of my friends and colleagues Thomas Walger, Vahid Babaei, Sergiu Gaman, Marjan Shapaski, Siavash Bigdeli, and Radhakrishna Achanta on my works. Their input during the development of the projects were very useful to enhance the results of this thesis.

Many thanks go to my friends Nikolaos Arvanitopoulos Darginis, Baran Gözcü, Bin Jin, Ioannis Klonatos, and Romain Rossier for their company during this journey. Finally, I am grateful to my family and my wife Maarja Arpa, who sincerely devoted their time motivating me during tough times.

*Lausanne, 15 November 2018*

Sami Arpa



# Abstract

Artists often take advantage of the limitations of the human visual system to create art that provides new experiences for the observers. Similarly, in this thesis, we explore new computational artistic compositions that create new visual experiences by relying on the limitations of the human visual system. We focus on two characteristics of the human visual system: One is the ambiguity that arises due to the differences between the eye projection space and the real physical space, and the second is the property of the integration of temporal information.

For the projection-space ambiguity, we propose two frameworks that enable two different artistic representations. First, we show how to create high reliefs that represent full 3D forms in a very limited space. The main challenges are to preserve the relationships of scene parts and the fine details of the shape. Second, we develop an approach for creating sculpture paintings where paintings and sculptures are blended within the same compositions. The main challenge is to ensure continuity between the 2D and 3D parts of the scene geometry. They create plausible effects for the viewer by showing a new relationship of 2D and 3D parts at each different angle.

For the temporal-space ambiguity, we propose a new stream of frames called *tempocode* that encodes the hidden information in video frames that show artistic dither matrices. In order to achieve this, we develop a visual-masking method that hides an input image, both spatially and temporally. Our masking function creates temporal and spatial variations on the frequency bands of the original input signal, with the constraint that the integrations of the final signal and the original signal are the same. The hidden information cannot be perceived by the human eye but it can be easily decoded by a long exposure photograph with a camera.





## Résumé

Les artistes profitent souvent des limites du système visuel humain pour créer des œuvres offrant de nouvelles expériences aux observateurs. De même, dans cette thèse, nous explorons de nouvelles compositions artistiques computationnelles qui créent de nouvelles expériences visuelles en nous appuyant sur les limites du système visuel humain. Nous nous concentrons sur deux caractéristiques du système visuel humain : la première est l'ambiguïté qui résulte des différences entre l'espace de projection oculaire et l'espace physique réel, et la seconde est la propriété de l'intégration de l'information temporelle.

Pour l'ambiguïté de l'espace-projection, nous proposons deux cadres qui permettent deux représentations artistiques différentes. Tout d'abord, nous montrons comment créer des hauts reliefs qui représentent des formes 3D complètes dans un espace très limité. Les principaux défis sont de préserver les relations entre les parties de la scène et les détails fins de la forme. Ensuite, nous développons une approche pour créer des peintures de sculpture où les peintures et les sculptures sont mélangées dans les mêmes compositions. Le principal défi est d'assurer la continuité entre les parties 2D et 3D de la géométrie de la scène. Elles créent des effets plausibles pour l'observateur, en montrant une nouvelle relation entre les pièces 2D et 3D à chaque angle différent.

Pour l'ambiguïté spatio-temporelle, nous proposons un nouveau flux d'images appelé *tempocode* qui encode l'information cachée dans des images vidéo montrant des matrices de vibrations artistiques. Pour y parvenir, nous développons une méthode de masquage visuel qui cache une image donnée, à la fois spatialement et temporellement. Notre fonction de masquage crée des variations temporelles et spatiales sur les bandes de fréquences du signal d'entrée original, avec la contrainte suivante : les intégrations du signal final et du signal original sont les mêmes. L'information cachée ne peut pas être perçue par l'œil humain mais elle peut être facilement décodée par une longue exposition photographique avec un appareil photo.



# Contents

<b>Acknowledgements</b>	<b>v</b>
<b>Abstract (English/Français)</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Non-photorealistic Rendering . . . . .	2
1.2 Sculpting and Reliefs . . . . .	3
1.3 Artistic Creations Relying on Optical Illusions . . . . .	5
1.4 Our Contributions in this Thesis . . . . .	6
<b>2 Background</b>	<b>9</b>
2.1 Spatial Perception and Representations of Forms . . . . .	9
2.1.1 Depth Perception . . . . .	9
2.1.2 2.5D Representations . . . . .	11
2.2 Temporal Perception and Approaches for Information Hiding . . . . .	15
2.2.1 HVS Perception . . . . .	15
2.2.2 Steganography and Watermarking . . . . .	16
2.3 Differential Coordinates . . . . .	18
<b>3 3D Representations in Limited Space: High Reliefs</b>	<b>21</b>
3.1 Introduction . . . . .	22
3.2 High Reliefs . . . . .	23
3.2.1 Forced Perspective . . . . .	24
3.2.2 Bring it to the Plane . . . . .	25
3.2.3 Compression and Artistic Control . . . . .	31
3.3 Results . . . . .	32
3.3.1 Choice of Methods. . . . .	32
3.3.2 Bas Relief Ambiguity. . . . .	33
3.3.3 Performance. . . . .	34
3.3.4 Limitations. . . . .	34
3.4 Conclusion . . . . .	35

## Contents

---

<b>4</b>	<b>Compositions Made of 2D and 3D: Sculpture Paintings</b>	<b>37</b>
4.1	Introduction . . . . .	37
4.2	Approach . . . . .	40
4.2.1	Extraction of ROI and Boundary . . . . .	41
4.2.2	Preparation of the Rendering Plane . . . . .	44
4.2.3	Attaching the ROI . . . . .	45
4.2.4	Incorporating Bas-Relief Profiles . . . . .	45
4.2.5	Textures . . . . .	46
4.3	Results and Discussion . . . . .	48
4.3.1	Lighting Artefacts . . . . .	48
4.3.2	Viewing Angle Ambiguity . . . . .	49
4.3.3	Scene Design and Cut Position . . . . .	50
4.3.4	Style Transfer . . . . .	50
4.3.5	Manufacturing . . . . .	50
4.4	Conclusion . . . . .	50
<b>5</b>	<b>Synthesis in the Temporal Domain: Tempocodes</b>	<b>53</b>
5.1	Introduction . . . . .	53
5.2	Self Masking . . . . .	54
5.3	Contrast Reduction for Masking Purposes . . . . .	57
5.4	Masking Functions . . . . .	57
5.4.1	Random Masking Function . . . . .	58
5.4.2	A Sinusoidal Composite Wave . . . . .	58
5.4.3	Temporal Dither Masking Function . . . . .	62
5.5	Averaging . . . . .	65
5.6	Results . . . . .	66
5.7	Conclusion . . . . .	67
<b>6</b>	<b>Conclusions</b>	<b>69</b>
<b>A</b>	<b>Appendix</b>	<b>71</b>
	<b>Bibliography</b>	<b>95</b>
	<b>Curriculum Vitae</b>	<b>97</b>

# 1 Introduction

Artistic synthesis has been a significant area of computer graphics with a focus on creating expressive styles by applying computational techniques. The computational methods for artistic synthesis are inspired by original artistic styles that can be found in paintings, sculptures, and drawings. Artists often take advantage of the limitations of the human visual system to create new styles and experiences for the observers. Similarly, in digital art, scientists might also take advantage of such limitations.



Figure 1.1 – Left: A painting from the Renaissance era, *An Old Man in Military Costume* (1630) by Rembrandt; Middle: *The Princesse de Broglie* (1853) by Jean Auguste Dominique Ingres; Right: *Femme au Chapeau* (1905) by Henri Matisse.

In computer graphics, establishing a sense of realism was the main concern at the early stages. After achieving, to a large extent, the goal of realism, artistic synthesis techniques [1] are gaining more attention. We make an analogy with the history of art. From the Renaissance period until the late 19th century, the focus of artists was to create photo-realistic paintings. Afterwards with the rise of oppositions to the traditional forms, many non photo-realistic art techniques were introduced, e.g. modernism leading to art movements such as impressionism, cubism, surrealism and fauvism (Fig. 1.1).

Although using computers for generating arts is quite a controversial issue in the discipline of fine arts, the majority of computer scientists in the field develop tools to help artists create art and to help art critiques in respect to aesthetic assessments. Other groups of scientists work to understand previous art works, styles, and movements, and to create algorithms that can mimic the original art pieces for the given input and user parameters. One of the main challenges of NPR is to understand how artists create imagery [2]. These questions are also of interest for the art historians working on similar matters. Interdisciplinary works between scientists and artists enable advancing both disciplines.

In this thesis, we present new computational tools for artistic synthesis. We do not deal with algorithms and statistical techniques that provide aesthetic assessments of existing artworks.

In the following, we will consider three different groups of artistic synthesis techniques in computer graphics; non-photo-realistic rendering, sculpting and reliefs, and optical illusions.

### 1.1 Non-photorealistic Rendering

Our synthesis techniques are not directly related to the non-photo realistic rendering (NPR). But we will shortly mention the NPR as it is the backbone of artistic synthesis in the computer graphics. The NPR is a field of computer graphics that focuses on creating rendering techniques that mimic original artistic styles [2]. Although early motivations for NPR were animated cartoons, e.g. "toon shading", today, NPR techniques are developed for different purposes such as painterly rendering, scientific visualization, architectural illustrations, and security applications [3].



Figure 1.2 – a: A cubist painting (Girl with mandolin (1910) - Pablo Picasso); b: Cubist style rendering generated from the photos [4]; c: Cubist style rendering from 3D geometry [5]. d: Neural network based cubist painting ([genekogan.com/works/style-transfer/](http://genekogan.com/works/style-transfer/)).

Many works were presented for painterly rendering. Starting from the pioneering works of Hertzmann [6], a majority of these works focus on creating realistic brushes. Some other works create styles according to specific art movements. One challenging example is cubism. A variety of methods are presented to generate the cubist style with quite different methods

such as 2D processing [4], 3D processing [5], and machine learning (Fig. 1.2).

Another important domain of NPR is pen-and-ink line drawing techniques. These techniques aim at representing forms for artistic, technical, and scientific illustration purposes [7].

The uncanny valley problem [8] in computer animations and games also increases the significance of the NPR. Although the trend was to create as realistic as possible animations for a few decades, the animation studios added an NPR layer to their pipeline after facing the uncanny valley problem. After creating a realistic animation, a final NPR rendering step is applied in order to give the animation a cartoon like look.

## 1.2 Sculpting and Reliefs

The 3D space is immense and its realistic representation in artificial forms, such as painting and sculpting, necessitates high-level artistic and technical know-how. Realistic paintings imitate the human visual system (HVS) to create strong depth sensations on 2D flat surfaces. The scenes are painted as one-to-one representations of their projected images on the retina. To mimic the retinal image, we know how our eye functions and what information is carried out from the retinal image to the brain. In the literature, the depth information is referred to by different means of optical relationships such as perspective, occlusion, and shading. They are called depth cues. By mimicking the depth cues, the forms can be represented in the limited space with the illusory depth effects.

In the sculpting discipline, a scene can be represented in three different forms [9]. The first form, called bas relief, is not very different from paintings, with respect to its construction method. The scene is similarly projected onto a 2D plane and the plane is slightly deformed to represent the changes in depth as elevations from the surface. The second form, high relief, is significantly different from paintings and bas reliefs. The original shapes of the scene are preserved instead of projecting them. But they are transformed into a limited space above a relief plane (Fig. 1.3). The third form is known as free standing sculpture. Free-standing sculptures are one-to-one imitations of parts of the original 3D scenes.

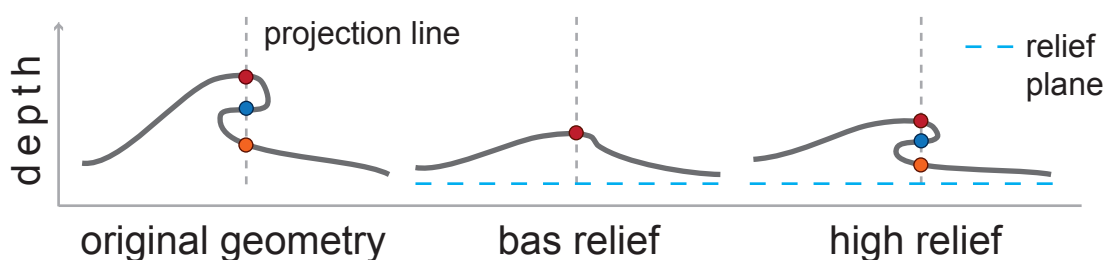


Figure 1.3 – Bas and high relief representations in 2D. High relief preserves original shape properties in limited depth space, whereas bas relief is the projection of the scene.

Free-standing sculptures do not need any enhancement for depth perception, whereas bas reliefs and high reliefs require special treatment, as limiting the space weakens depth percep-

tion. Several works address this problem by presenting computational methods for bas reliefs. The main approach is not too different from the one applied by painters. Pictorial depth cues are used on the projection of the scene for depth enhancement. However, these methods are not applicable for high-relief generation. There are two main challenges for high-relief generation:

- **The transformation of a scene into a limited space.** A transformation strategy should preserve the fine details of the figure, continuity of the overall surface, and the depth discontinuities within the scene.
- **Enhanced depth perception.** Limiting the depth space causes a loss of depth sensation, to a certain degree (Fig. 1.4). A new strategy should reinforce the sensation of depth by deforming the geometry.

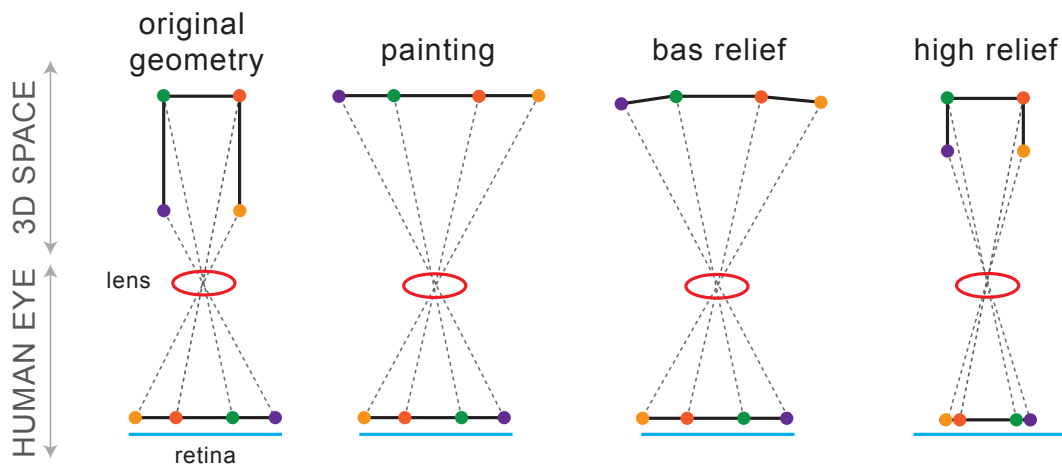


Figure 1.4 – Formulation of the retinal image for different representations. The painting and bas relief imitate the retinal image. The resulting retinal image provides a perception that is very similar to the original geometry. Without special provisions, high relief causes the loss of depth sensation, even though it preserves most of the properties of the original shape.

We address these two main challenges and present a method for synthesizing high reliefs in Chapter 3. Our method, based on the differential coordinates, brings the scene elements to the relief plane and preserves the depth discontinuities and surface continuities of the scene. We select an optimal number of attenuation points within the scene, attenuate these points towards the relief plane, and adapt new positions of all scene elements by using the differential coordinates. High-relief synthesis is 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. Furthermore, we show a perspective-deformation method that enhances the sensation of depth in the limited depth spaces.

We also present another sculpting form, sculpture paintings, in Chapter 4. This 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 this 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.

### 1.3 Artistic Creations Relying on Optical Illusions

Many artists use the limitations of the human visual system to create new artistic styles and new experiences for observers. Optical illusions have received much attention from the researchers working in cognitive science, neuroscience, and computer science as well.

Visual illusions can be created by exploiting variations of the visual response, according to the viewing conditions. The HVS exhibits variations in visual-contrast sensitivity depending on spatial frequency. Setlur and Gooch [10] use this fact to create facial images with different emotional states. These kinds of facial images rely on the CSF separation between peripheral vision and central vision [11, Ch. 5]. In a different way, Olivia et al. [12] in their “hybrid images” take advantage of visual sensitivity as a function of viewing distance. They decompose the luminance values of two images into frequency bands and, depending on the viewing distance, they choose different frequency bands for each image and combine them. In this manner, the resulting hybrid image has two different interpretations when it is viewed at different distances. A similar procedure is applied in the context of the disparity sensitivity function (DSF) in [13], which leads to two different depth percepts as a function of viewing distance, for a stereo 3D display.

Image mosaics [14] and collages [15] are other stylization methods using the limitations on the frequency perception. The icon-sized tile images are tightly packed into a larger container image, so that when they are seen together from a distance they represent this larger image. Complete photographs or their arbitrary-shaped cutouts (e.g., consistent with meaningful objects for the Arcimboldo-like effect) can be used as the tile images, whose details are readily visible from short distances.

Similar to mosaics, an artistic screening method is presented by Ostromoukhov and Hersch [16] in which artistic screen elements are used to generate halftones. Halftones are used to generate a global printed image. From a certain distance, the global image is perceived. Once observed closely or with a magnifier, micro-structures can be seen.

In another image fusion form, called camouflage images, features of a dominant image, especially its edges [17] and texture [18], are used to depict selected elements of a hidden image, whose recognition might require more effort due to sketchy presentation. Feature and conjunction searches are two different mechanisms of the HVS to recognize objects [Treisman and Gelade 1980]. The feature-search mechanism is fast and performs parallel processing by checking single features, whereas the conjunction-search mechanism is slow and interprets

multiple features to find objects. Chu et al. [2010] and Tong et al. [2011] use this separation to embed and hide objects into other percepts. They remove several features of the image, thus enabling fast recognition, and then they synthesize it with an apparent image. The hidden image remains imperceptible for a certain time.

Mitra et al. [19] generate emergence images of 3D objects, that locally appear as noise but become meaningful to the human observer when viewed as a whole and remain difficult to interpret by machines. In steganography a hidden message in the dominant image can be revealed using a decoder tool, such as a Cardan Grille or a Magic Lens [20], where a refractive lenslet array is placed over a seemingly unstructured image to reveal a hidden image. Copyright protection by watermarking is another example of hiding an ideally imperceptible image.

In a 2.5D representation, relief images [21] rely on a height-field surface, that, due to diffuse shading, depicts two unique images when illuminated from two directions. Similar to this work, Niloy et Pauly [22] create ambiguous 3D structures, where shadows of the object resemble the meaningful objects.

Some works take advantage of the HVS perception for the different illumination conditions. The sensitivity to luminance patterns of varying spatial frequencies can be characterized by the contrast sensitivity function (CSF). When the CSFs for scotopic and photopic vision are compared, over a tenfold reduction in the contrast sensitivity can be observed [23]. Also, the peak of sensitivity is shifted from  $8\text{cpd}$  in daylight vision to  $1\text{cpd}$  in night conditions. In tone mapping, visual acuity is typically modeled by low-pass filtering of the original image [24, 25, 26, 27, 28], where the cut-off frequency is a function of adaptation luminance as measured by Shlaer [29]. Arpa et al. [30] use the difference between night vision and day vision of the HVS. They conceive a combined image that is perceived as one image at night and as a different image during the day. The resulting images are artistically intriguing as well.

Our work presented in Ch. 5 is also a form of optical illusion. We take the advantage of limitations of the HVS in the temporal perception. The videos created by our framework consists of a hidden image which can only be observed with the conventional cameras. Underlying principles for the visual perception in this work is explained in Section 2.2.1.

### 1.4 Our Contributions in this Thesis

In this thesis, we present three new artistic synthesis techniques in Chapters 3, 4, and 5, that describe high reliefs, sculpture paintings, and tempocodes, respectively. All three works are completely new techniques rather than an enhancement of some previous synthesis approaches.

The high relief method (Ch. 3) describes how to create high reliefs computationally [33]. Although many works have been presented for the creation of bas reliefs, there were no works with a focus to produce high reliefs other than the initial relief paper in the field by Cignoni

[32]. But artistic realizations of high reliefs are as common as bas reliefs. We think our work was an important step in attracting further research in this domain. The main challenge of creating high reliefs consists of representing full volumes within a very limited space without dramatically deforming the original shape. We solve this problem with attenuation points and rely on differential coordinates to ensure the similarity of the deformed volume with the original one.

In the work of sculpture paintings (Ch. 4), we solve another completely new problem in computer graphics: the artistic compositions which has parts in 2D and other parts in 3D under the same continuum [31]. The sculpture paintings are not very common in artistic field such as reliefs. One of the reasons for this might be that it is very challenging to create such forms. The main challenge is to create a continuity between 2D and 3D parts by working on the texture and geometry. We develop a computational method for doing this. Our results can be perceived as nice in many different viewing directions and different illumination conditions.

Tempocode (Ch. 5) is our third synthesis work. We work in the temporal domain where artistic videos are created with our method [34]. The final video has two meanings. To the naked eye, the animation of artistic dither matrices is visible and the hidden image can be revealed with long exposure photography. In order to do that, we take the advantage of limitations of the human visual system in the temporal domain. In addition to their artistic nature, tempocodes can be used as a security feature at a certain time interval of videos like watermarks as a verification method for the authenticity.



## 2 Background

In this chapter, we give a background information related to our works presented in Chapter 3, 4, and 5. We first review how the depth perception is processed in the human visual system (HVS) and its relation to the different forms such as paintings, bas reliefs, and high reliefs. Then we show a computational method for bas-relief generation that creates plausible depth illusions. Next, we evaluate temporal perception of the HVS and mention some applications for information hiding. Finally, we evaluate differential geometry as it is used in all of our works that are explained in the next chapters.

### 2.1 Spatial Perception and Representations of Forms

In this section, we first investigate the depth perception based on different types of the depth cues explained by Palmer [35]. Then, we explain a state-of-the-art work by Weyrich et al. [36] to indicate how some of these depth cues are efficiently used to create illusory depth effects with bas reliefs. Their method is only valid for bas-relief generation because it is a projection-based method, where the scene is projected onto a 2D flat plane. To produce high reliefs, we manipulate the original scene geometry without projecting it.

#### 2.1.1 Depth Perception

Depth perception is the perceptual problem of recovering the distance to the objects in a 3D environment. The HVS operates an inverse problem in order to recover the depth information from the retinal image by using several ocular and optical depth cues. Some of these depth cues are given in Table 2.1 [35].

Ocular depth-cues come from the functionality of the eyes, the visual pathways, and interpretation by the brain. Accommodation and convergence are two main depth cues that come from ocular information. Accommodation indicates the focus of the lens, and convergence is the angle between the lines of the sight of the left and right eyes. These two cues are especially important in order to provide absolute depth information to the HVS. However, this informa-

Table 2.1 – Depth cues.

Depth cue	Binocular/ Monocular	Static/ Dynamic	Relative/ Absolute
Accommodation	monocular	static	absolute
Convergence	binocular	static	absolute
Binocular disparity	binocular	static	relative
Motion parallax	monocular	dynamic	relative
Convergence of parallels	monocular	static	relative
Position relative to horizon	monocular	static	relative
Relative size	monocular	static	relative
Familiar size	monocular	static	absolute
Texture gradients	monocular	static	relative
Shading and shadows	monocular	static	relative
Aerial perspective	monocular	static	relative

tion is only valid for objects that are not too far away. As a matter of fact, humans are not good at guessing absolute distances of far-away objects.

Optical depth-cues show information about the relation of the human eye with the 3D environment. Different than ocular depth-cues, most other optical depth-cues provide relative depth information. Once human eyes focus on an object, its absolute depth information comes with ocular depth-cues. The absolute distances of all other elements are calculated with the relative information coming from the optical depth cues. This could be considered as an integration operation in which the starting point is the absolute distance of the object being focused upon [35].

The information obtained from each eye could be slightly different. If a depth cue carries different information to the each eye, it is called a binocular depth-cue. Otherwise, it is named monocular depth-cue. Binocular disparity, which formulates the difference of the pictorial positions of the same objects on the left and right retinal images, is a significant binocular cue. This difference is zero for the object in the focus, and negative or positive for other objects. Binocular disparity provides significant information about the relative distance of the objects. Whereas, monocular cues are also sufficient for a strong depth perception. When we look at the environment with only one eye, we still have a sense of depth.

Although most depth cues are based on a static relationship between the observer and the environment, the dynamic information of this relationship creates another depth cue, named motion parallax. When the objects or the observer moves, the positions of the objects on the retinal image change in proportion to their distance to the fixation point of the eyes.

Artificial shapes such as paintings, bas reliefs, and high reliefs are capable of revealing only several depth cues. Fig. 2.1 shows a comparison of depth cues that differ in these three forms. Paintings are very powerful in terms of monocular depth cues. The convergence of lines within a perspective projection, as well as shading and shadows, are the strongest cues in paintings. However, dynamic-depth cues and binocular disparity do not exist in paintings. Similarly,

## 2.1. Spatial Perception and Representations of Forms

Retina	Depth cues	Painting	Bas relief	High relief
Image-based Representation	Shading and shadows	strong	weak	strong
	Dynamic shading and shadows	non-existent	weak	strong
	Binocular disparity	non-existent	weak	strong
	Motion parallax	non-existent	weak	strong
	Convergence of parallel lines	strong	strong	weak

Figure 2.1 – Information carried by depth cues varies depending on the artistic form (painting, bas-relief, and high relief).

bas reliefs also provide depth information with the convergence of parallel lines; and small elevations on the surface enable dynamic and binocular depth-cues for bas reliefs but they are weak. On the contrary, high reliefs reveal strong dynamic and binocular depth-cues because it preserves to some extent the 3D geometry. As the observer moves in front of a bas relief, shading, and shadows, and their relation to each other on the image representation change. Furthermore, a strong binocular disparity appears as the images on the left and right eye become slightly different. However, the convergence of parallel lines is weak in high reliefs (see Fig. 1.4). In Chapter 3, we describe our method for obtaining strong depth cues for high reliefs.

### 2.1.2 2.5D Representations

Bas relief is a sculpting technique that compresses shapes into a nearly-flat surface while maintaining the perception of the 3D scene from a single vantage point. In ancient cultures, it was a popular art form used for storytelling (Fig. 2.2). Today it is common in modern art, as well as in industrial design, architecture, and coin design.

The automatic generation of reliefs has recently become an important research topic. However, most works have focused on the design techniques for the creation of bas reliefs [37] rather than of high reliefs.

The dominating approach for designing a bas relief is based on a height field as initially presented in the work of Cignoni et al. [32]. 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 address mainly the trade-off between depth compression and the preservation of the details of the original 3D scene. Cignoni et al. [32] 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 away. Although this method produces acceptable bas reliefs for small compression ratios, it fails to preserve the details for scene elements with a wide range of depth. To solve this problem, several works

## Chapter 2. Background

---

[38, 36, 39] try to preserve most of the details from the scene by operating in the gradient domain [40] instead of the in depth map. Additionally, Weyrich et al. [36] apply a multi-scale approach for gradient compression. Some recent methods [41, 42] work both in the depth range and in the gradient domain to further enhance the resulting bas reliefs.

The main features of bas reliefs are

- **Depth.** The perspective foreshortening is the main depth-cue enables depth perception.
- **Object order.** The relative depth order of scene elements in the same local neighbourhood is preserved.
- **Depth discontinuities.** The depth discontinuities present in original 3D scenes are not maintained on the bas relief surfaces.

The algorithm of Weyrich et al. aims at maintaining these features on a nearly flat surface that gives the impression of a 3D scene when seen from a single vantage point. Their approach is based on tone-mapping algorithms where the goal is to map the contrast from the high-dynamic range of a scene to the low-dynamic range of a computer display [40]. In a similar manner as for bas reliefs, we want to map the original depth into a limited range for high reliefs.



Figure 2.2 – Bas reliefs. French (*left*) and Assyrian (*right*).

As discussed earlier in Ch. 1, bas reliefs are similar to paintings when considering that they are a projection of the scene (Fig. 1.3). Therefore, perspective projection, where parallel lines



## 2.1. Spatial Perception and Representations of Forms

---

converge, is the strongest depth cue for bas reliefs (see Fig. 2.1 and Fig. 2.3). The scene input to the algorithm is a perspective depth map of the 3D scene. It is as a heightfield  $h(x, y)$  to be used as a guide for deriving surface elevations. The heightfield is differentiated to perform all operations in the gradient domain. As the purpose is to reduce the dynamic range for depth, the gradient magnitudes are compressed, while the gradient directions  $\mathbf{v}(x, y) = \nabla h(x, y) / \|\nabla h(x, y)\|$  are kept constant. After gradient compression, new depth values  $h'(x, y)$  are recovered by integrating over the modified gradient  $\nabla g'(x, y)$ .

The depth discontinuities in the scene cause large gradients on the silhouettes. This results in depth discontinuities when the heightfield is integrated back to obtain the final relief. To remove these discontinuities, the gradient values on the silhouettes are set to zero, i.e. the heightfield is flat at the boundaries. The gradient magnitudes are modified in the following way:

$$s(x, y) = \begin{cases} C(\|\nabla h(x, y)\|), & 0 \leq \|\nabla h(x, y)\| < \vartheta_{sil}, \\ 0, & \vartheta_{sil} \leq \|\nabla h(x, y)\|. \end{cases} \quad (2.1)$$

where  $C$  is compression function and  $\vartheta_{sil}$  is silhouette threshold. The compression function  $C$  is a non-linear operator that preserves small gradients while attenuating large ones.

$$C(x) = \frac{1}{\alpha} \log(1 + \alpha x), \alpha > 0. \quad (2.2)$$

The parameter  $\alpha$  controls the amount of compression and is generally selected between 0.5 and 10.

Next, final gradient values are obtained by recombining the fixed orientations with the modified gradients:

$$\mathbf{g}'(x, y) = s(x, y)\mathbf{v}(x, y). \quad (2.3)$$

In order to recover the new heightfield  $h'(x, y)$ , an integration over the modified gradients  $\mathbf{g}'(x, y)$  is required. However, this vector field is not integrable. The following optimization function is used to find a solution with the least error:

$$h'(x, y) = \operatorname{argmin}_h \iint \|\nabla h(x, y) - \mathbf{g}'(x, y)\|^2 dx dy. \quad (2.4)$$

Similarly to Fattal et al. [40], this optimization is reformulated as a Poisson equation

$$\nabla^2 h = \text{div} g', \quad (2.5)$$

which is solved by using a general multi-grid solver.

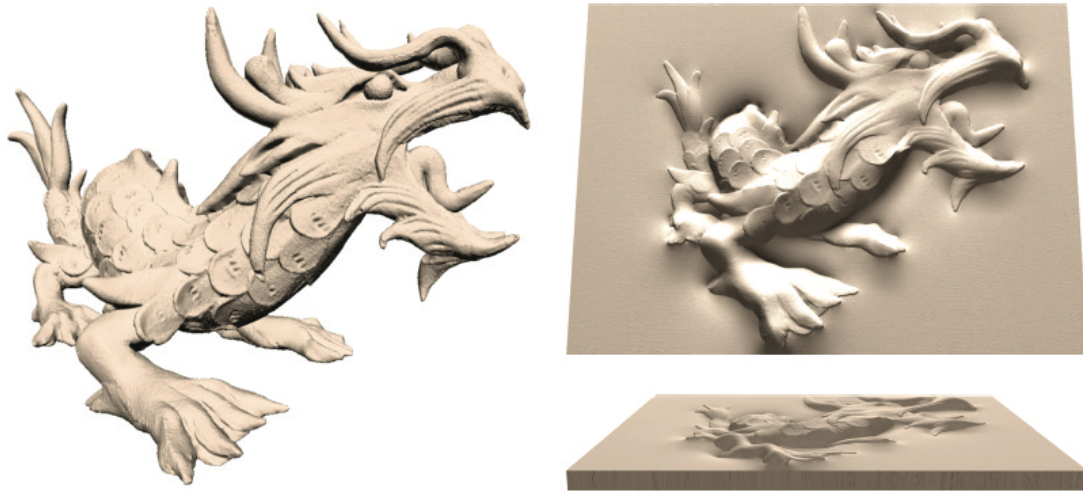


Figure 2.3 – Given a 3D input scene and a vantage point (*left*), a corresponding bas relief is generated (*right*) [36]. The depth effect is achieved with the perspective projection and by aligning the silhouette of the input with the relief plane. Courtesy [36].

The method of Weyrich et al. is one of the state of the art methods for digital bas-relief generation. As shown in Fig. 2.3, the results reveal plausible depth illusion for a single vantage point. However, this method is applicable only for bas-relief generation and does not apply for other forms such as high reliefs. In high reliefs, the scene geometry is preserved to a large extent. This enables some additional depth cues (see Fig. 2.1) and removes the limitation of a single vantage point for a plausible depth perception.

As state-of-the-art solutions for bas reliefs [37] are based on a height field, they are not adequate for generating high reliefs. As shown in Fig. 1.3, a bas relief can represent only a single point projected from the scene. In contrast, high reliefs might contain multiple points located on the same projection line. For the creation of high reliefs, Cignoni [32] 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 most complex 3D scenes. The authors also did 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. [43] introduces *appearance-mimicking surfaces* as a gener-

alization 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 that provides the desired appearance from a certain viewpoint. In contrast, our approach creates high reliefs that can be viewed from a wide range of viewing angles with an appearance similar to the original 3D shape as described in Chapter 3.

## 2.2 Temporal Perception and Approaches for Information Hiding

### 2.2.1 HVS Perception

Several works use the temporal domain to create artistic videos and animations. In this thesis (Ch. 5), we also present a method that takes advantage of the limitations of the human visual system to hide an image both spatially and temporally in the temporal domain.

We need to ensure the invisibility of the target image both spatially and temporally. For spatial invisibility, e.g., invisibility within a single frame, we utilize visual-masking methods. For temporal invisibility, e.g., when the video is playing, the temporal integration of the HVS is taken into account. Let us first describe spatial visual masking and then explain the temporal integration of the HVS.

**Visual Masking.** The presence of a mask can reduce the ability of a subject to detect a given target signal. This phenomenon is called visual masking [44]. A *masking threshold* is the limit of the contrast of a target for a given contrast of a mask. If the contrast of the target is higher than this threshold, masking is not achieved [45]. In classic image rendering, we mask the artifacts caused by operations such as compression, quantization, and dithering.

Visual masking is still not well-understood. Early channels of the HVS, where data is processed at the low level, and high level mechanisms of the HVS, exhibit different masking behaviours. For instance, masks having unstructured and structured shapes, respectively, result in completely different *masking thresholds* [46].

Masking is the strongest between stimuli located in the same perceptual channel. This is true both for early channels, e.g., stimuli having the same spatial frequencies and orientations (Fig. 2.4), and for higher-level mechanisms, e.g., stimuli with similar shape structures.

In order to implement visual masking, we generally need a *masking function*. A target signal can be masked with a mask signal by applying a contrast reduction on the target signal until the *masking threshold* is reached. As shown by Daly [47], for low-contrast masks, there is no masking. For high-contrast levels, as the mask contrast increases, the *masking threshold* increases in a non-linear fashion. However, an unstructured mask, e.g., a noise mask, and a structurally coherent mask, e.g., a sinusoidal grating, have different *masking threshold* curves [48].



Figure 2.4 – Visual masking. A uniformly laid out distortion is added to the reference image (left). The distortions are invisible on the vertical stripes of the zebra (right). Courtesy [49].

The *masking thresholds* also vary for different spatial frequencies and orientations. The human visual system exhibits different responses for a signal of the same contrast at different spatial frequencies and orientations. These responses are defined by a *contrast sensitivity function* (*csf*). Fig. 2.5 shows that the *csf* of the masked signal (Fig. 2.5b, Fig. 2.5d) is approximately proportional to the *csf* of the unmasked signal (Fig. 2.5a, Fig. 2.5c).

In order to explain visual masking, many other paradigms such as noise masking [50], pattern masking [51], and entropy masking [52] have been presented. Our approach is inspired by the *masking function* proposed by Daly. We reduce the contrast of the target in order to reach the *masking threshold* [47].

**Temporal Integration of the Human Eye.** Temporal integration is related to the ability of the HVS to sum the quantity of light exposed over time [53]. According to Bloch's law, the perceived luminous energy is a constant formed by the product of luminance and exposure time. For instance, the perceived luminance of a signal does not change if the intensity of the signal is halved, but the exposure time is doubled [54]. However, temporal integration might occur only for a critical duration. This critical duration is approximately  $40 \pm 10ms$  [53]. It depends on many factors such as spatial resolution and luminance adaptation, depending on photopic or scotopic vision. When the exposure time is longer than the critical duration, Bloch's law does not hold, and the perceived luminance depends only on the intensity of the signal.

With respect to our work, it is important that the temporal-averaging behaviour of the HVS does not reveal the parts of the target image that are to be hidden.

### 2.2.2 Steganography and Watermarking

Our synthesis work, the tempocode (Chapter 5), is related to the domain of steganography and watermarking as well. Therefore, we provide a short background about these domains.

**Steganography.** Steganography is a technique used for secret communication. The secret

## 2.2. Temporal Perception and Approaches for Information Hiding

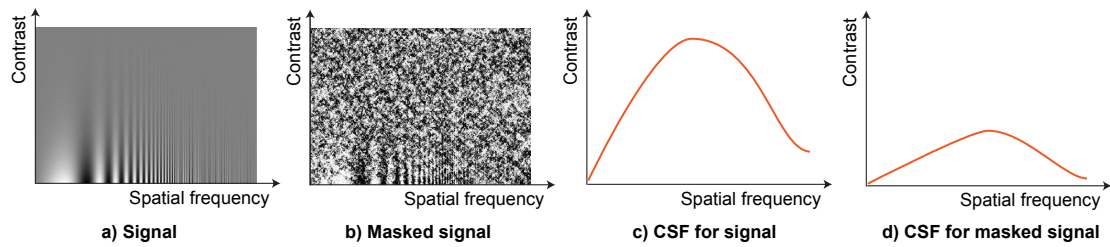


Figure 2.5 – Masking thresholds are proportional to the contrast-sensitivity function (CSF). a) The Campbell-Robson chart [55] indicates the shape of CSF function in an intuitive way. b) We apply our masking function on the Campbell- Robson chart. Due to masking, the visibility thresholds are reduced. This reduction (d) is fairly proportional to the contrast-sensitivity function (c) as asserted by [47].

message is not necessarily related to the visible content. Furthermore, the secret information is more important than the visible content. The main concern of steganography is the undetectability of the hidden content [56]. Many steganography methods act on the spatial domain of the image [57, 58, 59]. In most of these methods, the hidden content is embedded by changing the least significant bits of pixel values. Embedding at the spatial level is sufficient to deceive the HVS. However, the resistance of these methods to attacks are weak. More advanced methods in steganography use the spatial-frequency domain where embedding is performed at the cosine-transform level [60, 61, 62]. McKeon [63] shows how to use the discrete Fourier transform for steganography in movies. Some adaptive steganography methods consider the statistics of image features before embedding the message [64, 65]. For example, the noisy parts of an image are more suitable for embedding than the smooth parts [66].

**Digital Watermarking.** Digital watermarking is used for the protection of digital content. Different than steganography, the visible content is more important than the hidden content. The strength of watermarking methods is related to the difficulty of removing the hidden content from the visible content. The watermark marks the digital content with an ownership tag. Copyright protection, fingerprinting to trace the source of illegal copies, and broadcast monitoring are the main purposes of digital watermarking [67]. In reversible watermarking techniques, a complete restoration of the visible content is possible with the extraction of the watermark [68]. Several approaches use lossless image compression techniques to create space for the watermarking data [69, 70, 71]. Other work is based on histogram modification [72, 59, 73] and quantization algorithms [68] to embed watermark data. Although many different algorithms are used, the main goal of all reversible watermarking methods is the same: to avoid damaging sensitive information in the visible content and to enable a full extraction of the watermark and original data. For the extraction of the watermark, a retrieval function is required. Complex embedding functions result in complex retrieval functions, which requires special software. This is one of the disadvantages of digital-watermarking techniques. Although they provide a high level of security, the authenticity cannot be controlled as easily as in traditional security-printing techniques [3]. In many security-printing techniques, the originality of the content can be verified by humans with simple tools such as a magnifier, a

directed light, or a UV light.

Our method does not compete with watermarking or with conventional steganography. Visual watermarking protects the content of an original video. Whereas, we generate synthetic signals to hide visual information. Most steganographic methods use very complex decoding procedures. But, our method reveals information without using any decoding algorithm, i.e. by long exposure photography. Therefore, our method does not compete with existing visual watermarking or steganography methods.

### 2.3 Differential Coordinates

In this thesis, we use several mathematical models. In the next three chapters, we use differential coordinates. Therefore, we give an overview of the differential coordinates here.

The representation of 3D surfaces in differential coordinates, also known as Laplacian coordinates, was previously described in [74, 75]. 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 [76].

Surface representation is a significant topic in computer graphics. The way a 3D scene is defined determines the type of operations that can be done on the 3D scene. The traditional Cartesian coordinates do not carry any information about local shape properties such as the size and orientation of local mesh elements. Consequently, undesired deformations occur when operations such as mesh editing, compression, and morphing are performed by using Cartesian coordinates. Sorkine [76] describes alternative representations based on differential geometry. In contrast to Cartesian coordinates, differential geometry provides information about the local characteristics of the shape. This enables to apply various mesh processing operations without losing the essence of original geometry.

Cartesian coordinates are transformed into differential coordinates for a differential geometry representation. Let  $\mathbf{v}_i = (x_i, y_i, z_i)$  denote Cartesian coordinates of a vertex of a scene  $M$ . The differential coordinates  $\delta_i$  of vertex  $\mathbf{v}_i$  are the difference between absolute coordinates of  $\mathbf{v}_i$  and the center of mass of its one-ring neighbours.

$$\delta_i = \mathbf{v}_i - \frac{1}{d_i} \sum_{j \in N(i)} \mathbf{v}_j, \quad (2.6)$$

where  $d_i$  is the number of connected neighbour vertex  $\mathbf{v}_j$  and  $N$  is the set of neighbour vertices that are connected to  $\mathbf{v}_i$  with an edge. Alternatively, the differential coordinates can be calculated as a sum of weighted differences. The weights can be calculated according to the area of corresponding triangles in order to preserve the area or according to the cotangent values of corresponding angles in order to preserve the angles in the reconstruction (see

Appendix III).

This transformation can be represented in matrix form. Let  $\mathbf{A}$  be the adjacency matrix of the scene  $M$  such that:

$$A_{ij} = \begin{cases} 1, & (i, j) \text{ is connected} \\ 0, & \text{otherwise,} \end{cases} \quad (2.7)$$

and let  $\mathbf{D}$  denote the diagonal degree matrix where  $\mathbf{D}_{ii} = d_i$ . Then, the symmetric Laplacian matrix, which is used to transform from Cartesian coordinates to differential coordinates, becomes  $\mathbf{L} = \mathbf{D} - \mathbf{A}$ . Let  $\mathbf{V}$  be  $n \times 3$  matrix which contains Cartesian coordinates of all  $n$  vertices in the scene  $M$  and let  $n \times 3$  matrix  $\Delta\mathbf{V}$  contain corresponding differential coordinates. The transformation is as follows:

$$\Delta\mathbf{V} = \mathbf{L}\mathbf{V} \quad (2.8)$$

An inverse solution,  $\mathbf{V} = \mathbf{L}^{-1}\Delta\mathbf{V}$ , is required to reconstruct the surface from differential coordinates. The inverse solution is undefined because  $\mathbf{L}$  is singular hence non-invertible. If all vertices are connected, then the rank of Laplacian matrix becomes  $n - 1$ . We need to solve a full-rank linear system to find a unique solution. Therefore, it is required that the Cartesian coordinates of at least one vertex  $i$  are specified as the solution by substituting 0 to all elements on  $i$ th row and 1 for the element  $(i, i)$  in Laplacian matrix  $\mathbf{L}$ . Then  $\mathbf{L}$  becomes invertible and a unique solution can be constructed from differential coordinates. The number of substitutions will increase the number of constraints. If the constraints are directly substituted by the Laplacian matrix, the system can be solved with square matrix solvers such as LU decomposition. Alternatively, we can add the positional constraints as soft constraints by putting the equations as additional rows of the linear system. Then, the system can be solved in the least-squares sense e.g. by QR decomposition.

The Laplacian matrix also forms a good basis for spectral-mesh representation. Once an eigen decomposition is applied on the Laplacian matrix, its eigenvectors correspond to different frequency bands. For instance, when eigenvectors are ordered according to the eigenvalues  $\lambda_i$  such that

$$0 = \lambda_1 < \lambda_2 < \lambda_3 \dots < \lambda_n, \quad (2.9)$$

the first eigenvectors with the lowest eigenvalues correspond to a smoothly varying function

on the mesh, the last eigenvectors represent a function with higher frequencies, as shown on Fig. 2.6. These functions that correspond to the different frequency bands enable to apply efficient operations, such as compression and smoothing, in a way that is similar to operating for image compression in Fourier domain. In order to smooth the surface, low frequencies can be compressed or high frequencies can be eliminated.

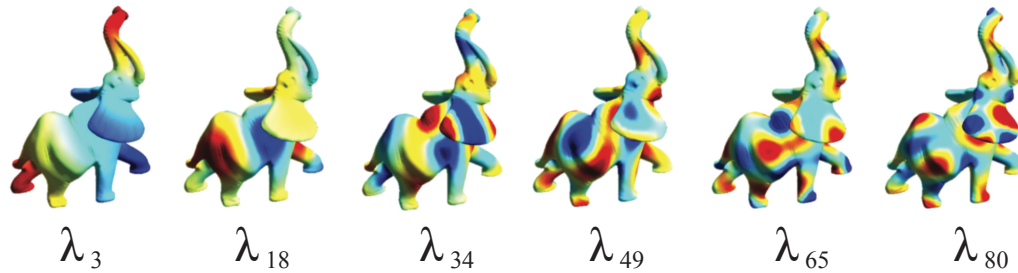


Figure 2.6 – Eigenvectors, which are ordered according to eigenvalues  $\lambda$ , correspond to the functions of different frequency bands. The colors of vertices are assigned proportional to the variation on the corresponding eigenvector.

Differential coordinates and spectral mesh processing enables very efficient mesh processing for various goals. Interactive mesh editing can be achieved by using a Laplacian matrix, once it is decomposed with Cholesky factorization. Efficient shape representations and compression operations can be performed. Furthermore, differential geometry is efficiently applied for highpass quantization, mesh watermarking, and shape interpolation.

Gradient domain representation provides a powerful tool for creating prominent bas reliefs. Similarly, the domain of differential geometry can be used to generate high reliefs. The purpose is to transform scene elements within a limited space while maintaining its main characteristics. In the next chapter, we propose our high-relief method that is based on differential geometry.



### 3 3D Representations in Limited Space: High Reliefs

In this chapter, we present a method for synthesizing high reliefs. 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, as mentioned previously in Ch. 2. 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.

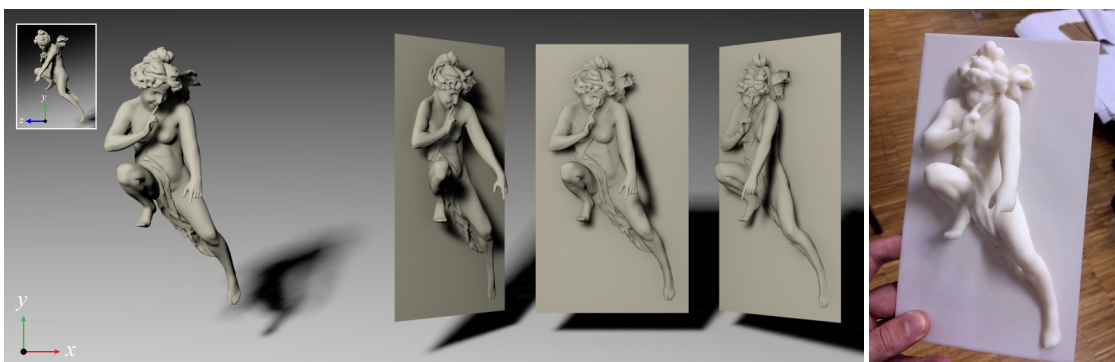


Figure 3.1 – With an original 3D scene (*left*), we generate a high relief (*middle*) that incorporates free-standing sculpture elements located within a limited depth range. Today, high reliefs can be produced thanks to additive fabrication technologies such as 3D printing (*right*).

### 3.1 Introduction

High relief, also referred to as *alto-rilievo*, is a sculpting technique in the plastic arts [77]. 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. 3.2) as shown in Section 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 by 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.

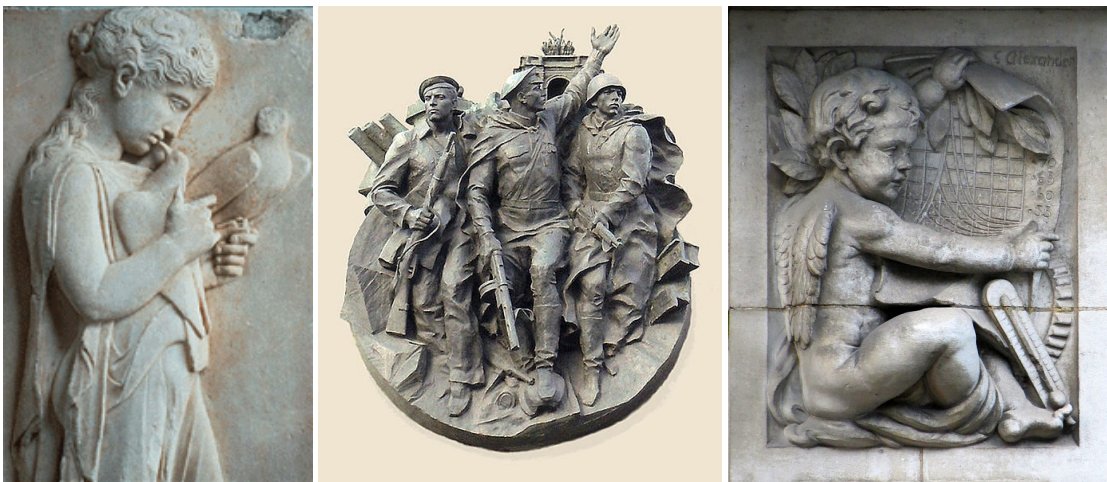


Figure 3.2 – High relief examples. *Left*: Greek relief in marble, Girl with Doves, 300 B.C.; *Middle*: Hero-City Obelisk, St. Petersburg, 1985; *Right*: Cherub sculpture in high relief.

Although techniques have lately been proposed for the design of bas reliefs [37], there have not been many attempts to address the challenges of high reliefs. The fundamental challenges for high-relief design are the following [78]:

- **Depth discontinuities.** Parts of the scene are detached from the relief plane and result in depth discontinuities (Fig. 1.3). Although 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 a different perception of the model 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 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 [74, 75]. The continuity and fine details of the original scene and the depth discontinuities of the high-surface parts are preserved (Fig. 3.1). 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.

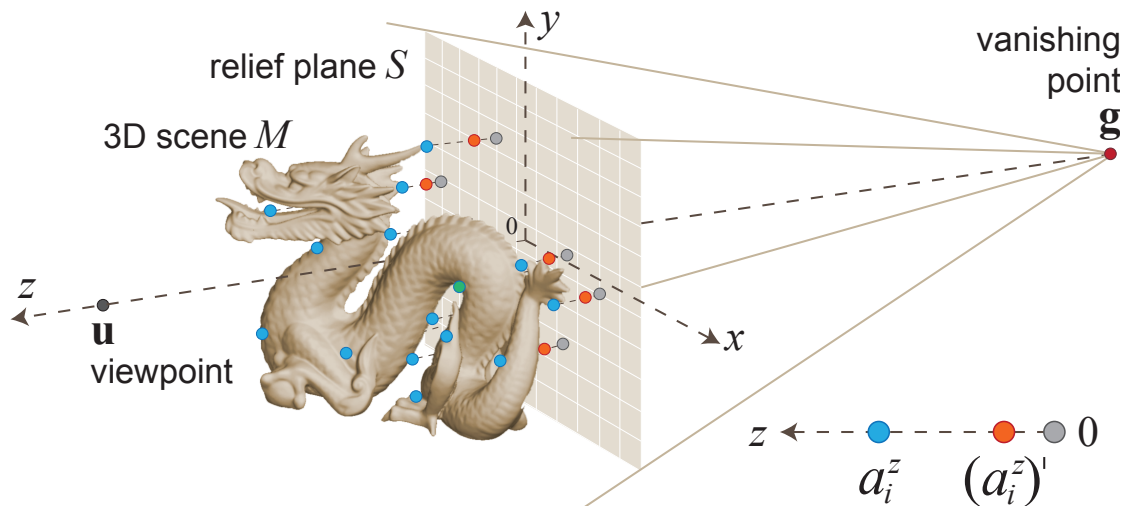


Figure 3.3 – A scene  $M$  with viewpoint  $\mathbf{u}$ , relief plane  $S$ , vanishing point  $\mathbf{g}$ , and  $k = 16$  attenuation points.

## 3.2 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 thus preserved.

Let us introduce the notations used hereinafter in this chapter. We consider a relief plane  $S \in R^3$  and a 3D scene  $M = \{V, \Delta V\}$ , where  $V = \{\mathbf{v}_1, \dots, \mathbf{v}_n\} \subset R^3$  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  $\Delta V = \{\Delta \mathbf{v}_1, \dots, \Delta \mathbf{v}_n\} \subset R^3$  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  $\mathbf{u}$  and the vanishing point  $\mathbf{g}$ : they guide the perspective deformation. We partition the scene into  $k$  clusters  $C = \{c_1, \dots, c_k\}$ . 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. 3.3 shows the scene, the viewpoint, the relief plane, and the vanishing point within a coordinate system centered in the relief plane.

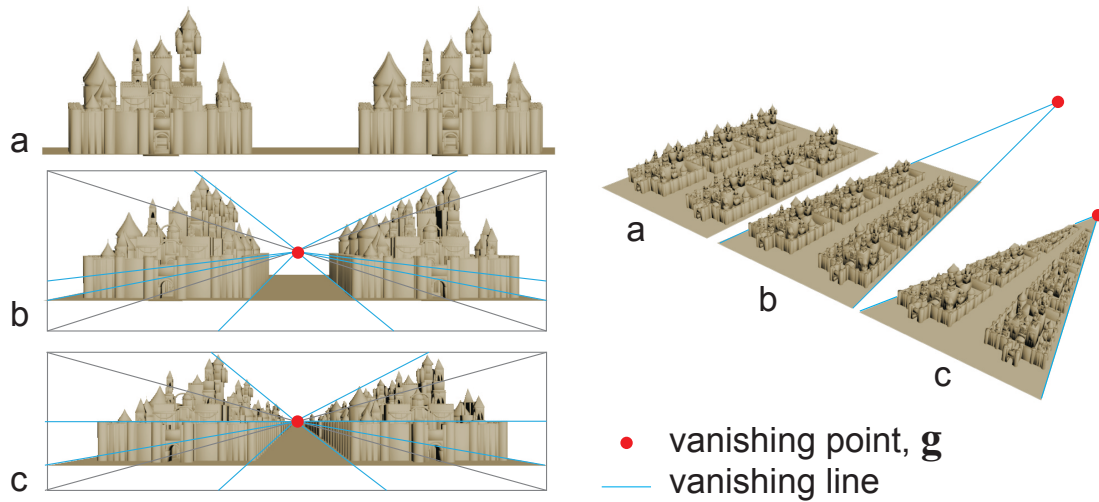


Figure 3.4 – Perspective deformation. *a*: Original input scene; *b*: the scene deformed for a far vanishing point; *c*: the scene deformed for a near vanishing point.

### 3.2.1 Forced Perspective

Perspective foreshortening [79] is a key tool for achieving a depth illusion when projecting 3D scenes onto a 2D plane. However, it is not applicable for high reliefs, because we do not project 3D points onto a 2D plane. Nevertheless, as we limit the depth range, deforming the geometry along vanishing lines is useful for enhancing 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 [80, 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 and Fig. 3.5. The deformation is controlled with the vanishing point  $\mathbf{g} = [g^x \ g^y \ g^z]$  located behind the relief plane and the viewpoint  $\mathbf{u}$ .

Choosing  $\mathbf{g}$  at infinity ( $g^z = \infty$ ) does not cause any deformation of  $M$ , whereas choosing it closer to the relief plane  $S$  creates a corresponding perspective deformation (Fig. 3.4). Note that as 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.

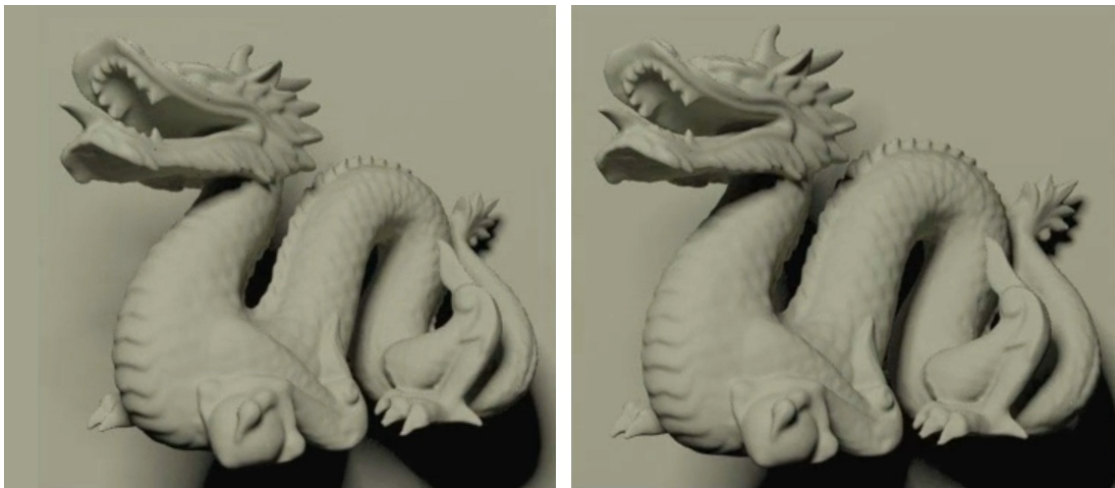


Figure 3.5 – *Left*: high relief with perspective deformation; *Right*: high relief without perspective deformation.

### 3.2.2 Bring it to the Plane

Instead of directly compressing the depth range, we bring scene elements toward the relief plane  $S$  and attach some of them to the plane. Yet we 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  $\Delta V$  of the vertices.

**Attenuation points.** The selection of attenuation points determines to a large extent the final high-relief geometry. Our goal is to select these 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 a small number of attenuation points  $k$  in order to avoid excessive deformations.
- Preserve the original depth order of all scene vertices in local neighbourhoods.

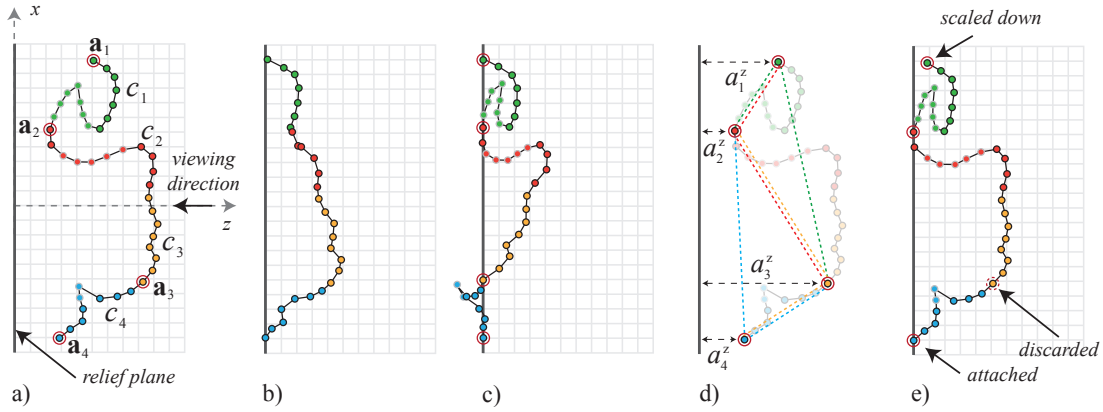


Figure 3.6 – Clustering and selection of attenuation points on a simple curve in 2D. Each circle indicates a vertex. Vertices with dark borders are visible vertices and vertices with light borders are invisible vertices. The vertices having the same color belong to the same cluster. a) The vertices are 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. [36]. 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.

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 [81] 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 might 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 in 2D more easily than in 3D. Fig. 3.6a 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. 3.6c, 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. 3.6c). 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. As the angles between attenuation points are not significant, instead of using cotangent weights as described in Appendix III, we use uniform weights:

$$\Delta a_i^z = a_i^z - \frac{1}{|N|} \sum_{j \in N} a_j^z \quad (3.1)$$

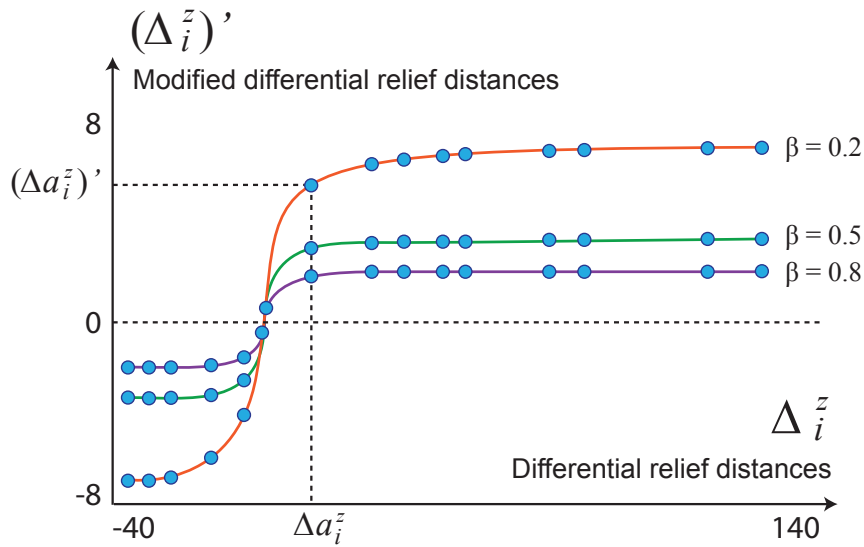


Figure 3.7 – Modified differential-relief distances for  $k = 16$  attenuation points after applying the attenuation function proposed by Bian et al. [42] (see Eq. 3.3) for the dragon model in Fig. 3.3.

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:

$$\rho_i = \frac{\Delta a_i^z - \Delta a_{min}^z}{\Delta a_{max}^z - \Delta a_{min}^z} \quad (3.2)$$

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

where  $t_1$  is the threshold for discarding an attenuation point and  $t_2$  is the attachment threshold.

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

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  $\rho_i$ , 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, as it might cause excessive deformations (see  $\mathbf{a}'_3$  and  $c_4$  in Fig. 3.6c). In contrast, a small  $\rho_i$ , which corresponds to an attenuation point with 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 attached to the relief plane (see  $\mathbf{a}'_2$  and  $\mathbf{a}'_4$  in Fig. 3.6d). 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  $\Delta a_i^z$  by using an attenuation function proposed by Bian et al. [42].

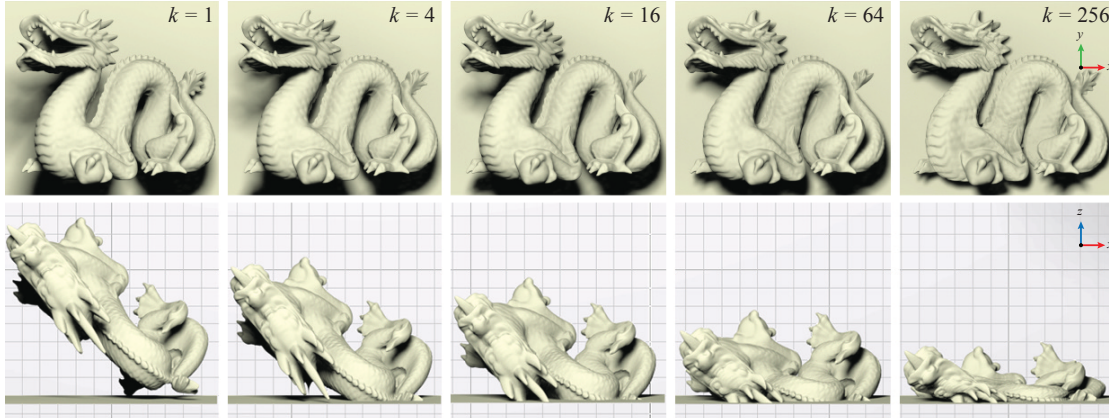


Figure 3.8 – Increasing the number  $k$  of attenuation points increases the number of scene elements that are brought to the relief plane and decreases the overall depth range.

$$(\Delta a_i^z)' = \frac{\arctan \beta_i \Delta a_i^z}{\beta_i} \tag{3.3}$$

We use attenuation factor  $\beta_i = 0.2$  for all attenuation points if not stated otherwise. As shown in Fig. 3.7, this affects large differential values very strongly. A wide range of differential values is almost equalized [37].

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:



$$(a_i^z)' = 0, \quad i \in \{b, \dots, k\}, \quad b < k \quad (3.4)$$

where  $\{b, \dots, k\}$  are the indices of the attachment points. Under these constraints, we find the distances of the other attenuation points  $(a_i^z)', \quad i \in \{1, \dots, b-1\}$  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. [76] by minimizing the following cost function:

$$E((a_1^z)', \dots, (a_k^z)') = \sum_{i=1}^k \|(\Delta a_i^z)' - \Delta(a_i^z)'\|^2 + \sum_{i=b}^k \|(a_i^z)'\|^2 \quad (3.5)$$

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 system of equations [76] (see Appendix IV). Fig. 3.6e 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  $\Delta V$  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 [82] (see Appendix III). Next, we reconstruct the surface by fixing the new positions of the attenuation points:

$$\mathbf{v}'_i = \mathbf{a}'_i, \quad i \in \{m, \dots, n\}, \quad m < n \quad (3.6)$$

where  $\{m, \dots, n\}$  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  $\mathbf{v}'_i, \quad i \in \{1, \dots, m-1\}$  using the differential values  $\Delta \mathbf{v}'_i, \quad i \in \{1, \dots, m-1\}$ . We minimize the square differences between the differential coordinates of new vertex positions  $\Delta \mathbf{v}'_i$  and the original differential coordinates  $\Delta \mathbf{v}_i$  for all vertices, except the attenuation points.

$$E(V') = \sum_{i=1}^{m-1} \|\Delta \mathbf{v}'_i - \Delta \mathbf{v}_i\|^2 \quad (3.7)$$

This optimization problem is solved as a sparse linear system of equations (see Appendix V).

Instead of treating the constraints in the least-squares sense, as suggested by Sorkine et al. [76], we substitute them into the linear system. Then, the positions of attenuation points are completely preserved.

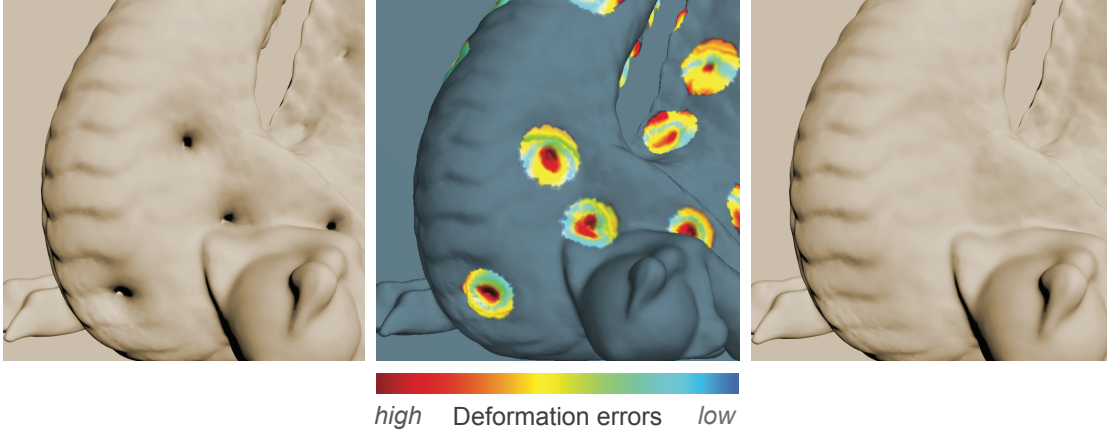


Figure 3.9 – Undesired deformations become more visible as the number  $k$  of attenuation points is increased (*left*). We detect local deformation errors by comparing the depth orders (*middle*) and apply a reconstruction to correct the errors (*right*).

**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. 3.8). However, having sparsely distributed positional constraints results in undesired deformation errors. As we apply a least square solution, errors are locally distributed around the attenuation points (Fig. 3.9). The use 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 Appendix VI and VII. Instead of using these methods, 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  $\epsilon$  is calculated for a vertex  $\mathbf{v}$  in the neighbourhood of  $\mathbf{a}_i$ :

$$\epsilon = \frac{|v'_o - v_o|}{p} \quad (3.8)$$

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 that are around attenuation points and assign  $\epsilon = 0$  for all other vertices. Next, we reconstruct the surface parts that have high errors. All vertices with error  $\epsilon < t_3$  are fixed as positional constraints. We use  $t_3 = 0.2$ . This value was 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. 3.9, with this strategy, we remove local errors without relaxing the depth range or introducing new deformations.

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

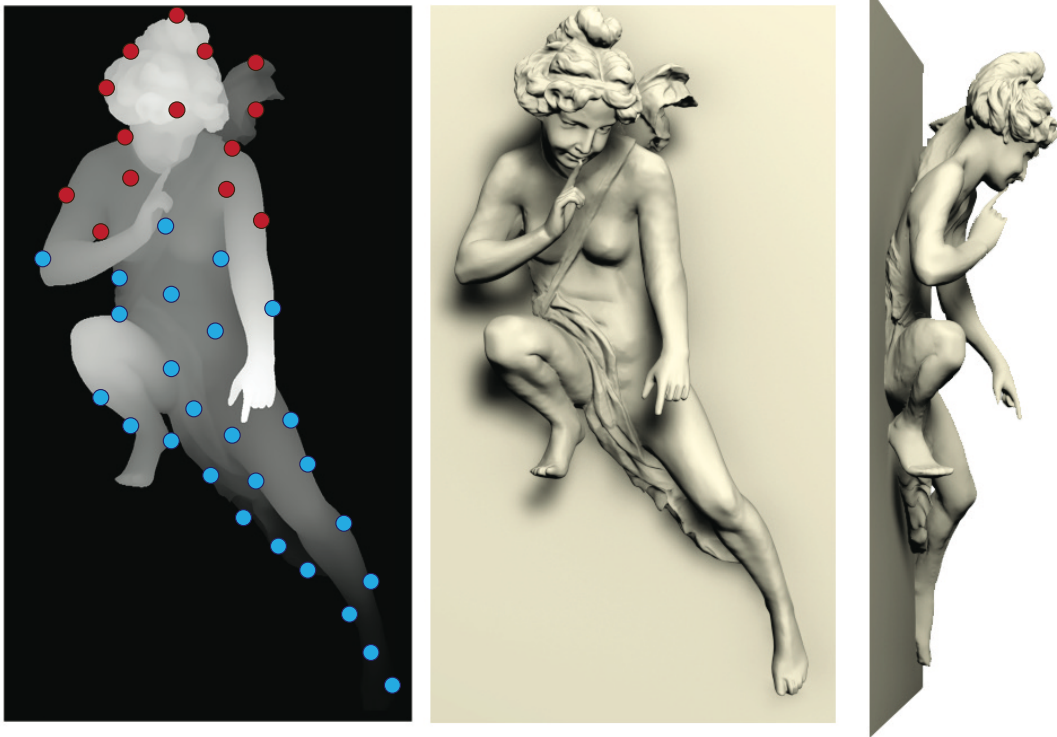


Figure 3.10 – The users can control the attenuation factor  $\beta_i$  for each attenuation point and possibly release attachment points. The red dots indicate the attenuation points with  $\beta_i = 0.1$  and the blue ones with  $\beta_i = 0.8$ . Smaller  $\beta_i$  values enable scene elements to be detached from the relief plane.

### 3.2.3 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. 3.8, but more undesired deformations might occur. In these cases, a range-compression method might be preferred for further compression. We apply a range compression similar to the method of Cignoni et al. [32], where the vertices close 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  $\alpha_{max}$  and minimal  $\alpha_{min}$  compression ratio. A local compression value  $\alpha_i$  is calculated for each vertex  $v_i$  by mapping the range of relief distances  $(v_i^z)'$  in the interval between  $\alpha_{min}$  and  $\alpha_{max}$ . Finally, the relief distance  $(v_i^z)'$  of a vertex is multiplied by its local compression ratio

$\alpha_i$ . For the examples in this chapter, we only indicate  $\alpha_{max}$  by referring to it as  $\alpha$  and use  $\alpha_{min} = \alpha_{max}/2$ .

As shown in Fig. 3.11 and Fig. 3.13 as well as in the Appendix VII, we obtain different results depending on the selection of  $k$ ,  $\alpha$ ,  $\beta$ , and vanishing point  $\mathbf{g}$ . For the artistic control of the resulting high relief, these parameters can be set by the designers. They can also define the attenuation factors  $\beta_i$  for the attenuation points, see Eq. 3.3. Changing  $\beta_i$  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. 3.10).

### 3.3 Results

We discuss the results and illustrate potential uses based on different settings. As discussed in Sec. 3, we provide artistic control for the number of attenuation points  $k$ , the depth-compression value  $\alpha$ , as well as the attenuation factor  $\beta$ , 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 with bas reliefs, the preservation of depth discontinuities yields strong shadows. Fig. 3.11 shows a comparison between a simple compression of the depth range ( $k = 1$ ,  $\alpha = 0.3$ ) and our method 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  $\alpha$  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. 3.13 and the additional results for different type of scenes are shown in Appendix VII.

#### 3.3.1 Choice of Methods.

We use different solutions for finding the new positions of the attenuation points (Eq. 4.3) and of all vertices (Eq. 3.7). As shown by Sorkine et al. [76], using the least-squared method is preferable in most scenarios. However, in the construction step (Eq. 3.7), 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 Appendix VII). However, when we fix the attenuation points, the artifacts remain localized around the attenuation points and are easily detected (Fig. 3.9).



Figure 3.11 – 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*:  $k = 1$ ,  $\alpha = 0.3$ ; *d*: *close-up*), the details are lost and the scene is not well attached to the relief plane (*d*). With more attenuation points (*a*:  $k = 128$ , *no compression*; *e*: *close-up*), the scene elements are brought to the relief plane without inducing a loss of details.

### 3.3.2 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

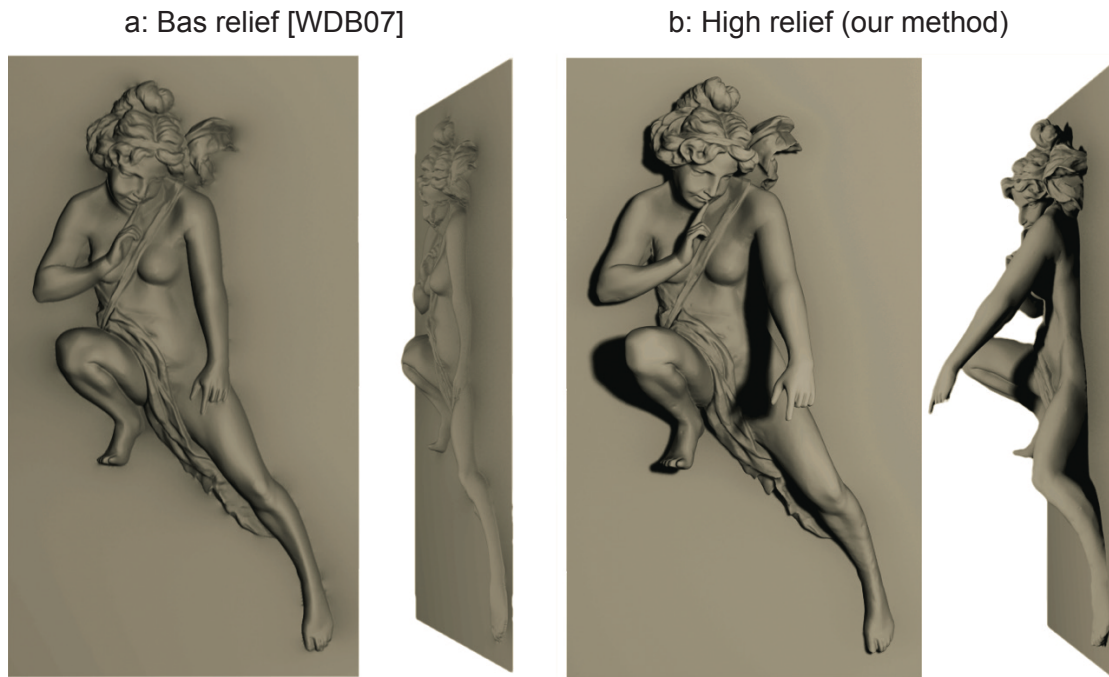


Figure 3.12 – The comparison of a bas relief and a high relief at different viewing angles under the same lighting conditions.

and with a correct viewpoint. This is called "bas relief ambiguity" [83]. 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. 3.12). 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. 3.12 shows a comparison of our high relief with a bas relief generated with the method of Weyrich et al. [36]. The appearance degrades when viewing the bas-relief from the side, whereas for the high relief it is kept intact.

### 3.3.3 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 that takes 34 seconds. The execution times were measured on a computer with an Intel Xeon E3-1225 processor.

### 3.3.4 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 for keeping all visible parts of the scene in front of the relief plane. For complex scenes, having the parts of scenes that are not directly connected to each other

can cause the inter-penetration of these parts, in the case of too high constraints induced by a large number of attenuation points.

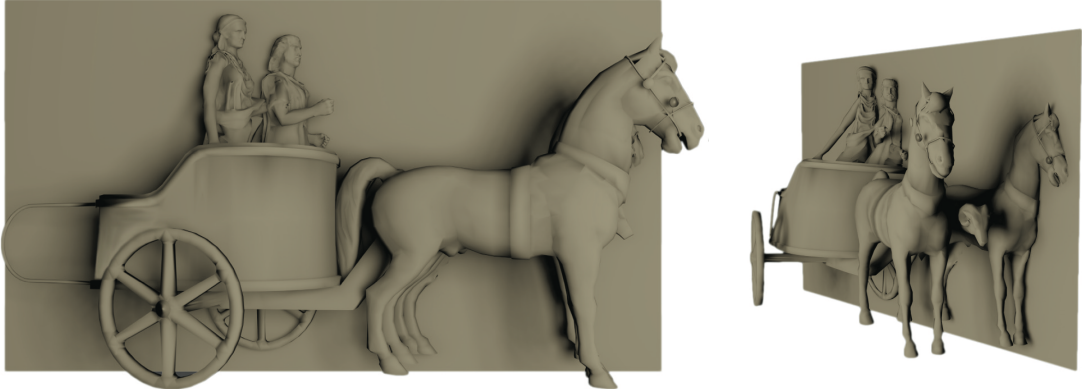
### 3.4 Conclusion

We have presented a method for creating visually pleasing high reliefs from 3D scenes that is simple to operate and still permits 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 freedom to artists to control the parameters that adjust the depth range, the attachment to the relief plane, and the range compression. Our proposed high-relief synthesizing method can 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.

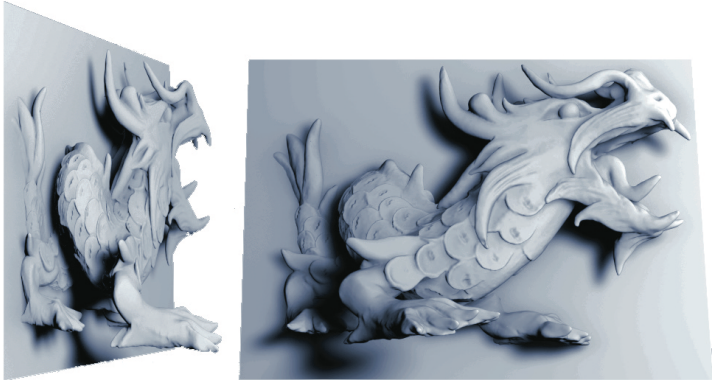
Our high-relief method is a first step towards solving the problem of automatic generation of high reliefs and it can inspire future works. The method can be extended to create high reliefs on arbitrary-relief surfaces. Optical illusions can be created by considering light changes, shadow effects, and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

### Acknowledgements

The 3D models used in this chapter and in the Appendix are taken from the following resources: Dragon, Asian Dragon, Happy Buddha, and Armadillo from the Stanford 3D scanning repository; other models from <http://www.archive3d.net>.



a:  $k = 128, \alpha = 1.0$



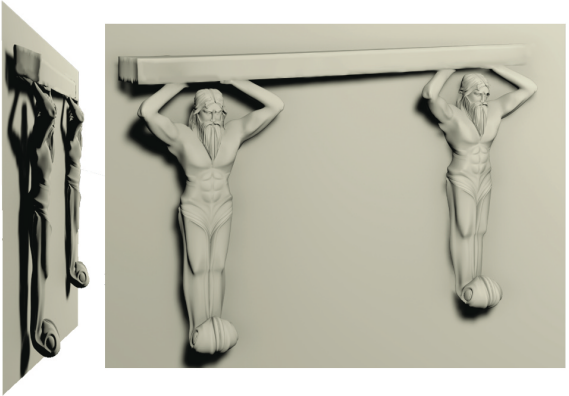
c:  $k = 128, \alpha = 0.8$



d:  $k = 128, \alpha = 0.8$



b:  $k = 256, \alpha = 0.8$



e:  $k = 128, \alpha = 0.8$

Figure 3.13 – High-relief examples for different type of scenes.



## 4 Compositions Made of 2D and 3D: Sculpture Paintings

In this chapter, we present a framework for automatically creating a type of artwork in which 2D and 3D contents are combined 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. When 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 ensuring 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 boundary 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.

### 4.1 Introduction

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, whereas others emerge from the 2D plane as 3D objects, as shown in Fig. 4.2. This creates interesting visual effects for observers.

Artists who blend a painting and a sculpture must meet several challenges. First, the visual information of a painting does not fundamentally change as a function of the viewing position, whereas a sculpture reveals new information at each viewing angle. In 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 the 3D elements. Depending on the type of light and its position, the



Figure 4.1 – 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

diffuse shading, specular highlights, and shadows can change to a great extent the appearance of the 3D elements. Contradictions with the 2D part can arise.

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

The initial challenge consists in 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, as they rely on elevation from the surface. Instead, we cut the region of interest (ROI) directly from the scene, create a grid that has homeomorphic boundaries and a similar topology as the 3D part; then we attach it to the grid. A simple cut operation by intersecting



Figure 4.2 – A sculpture painting created by the artist Marshall Mithourad.

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 that looks smooth on both the camera direction and the 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 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 but also ensure consistency between the appearance properties of the flat part and the 3D part, especially in case of specularities. Furthermore, we define pre-lighting conditions for baking the textures of the scenes, and we perform a style transfer [84] 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. 4.3). 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 incorporates 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 of the represented object 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. 4.1) 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.

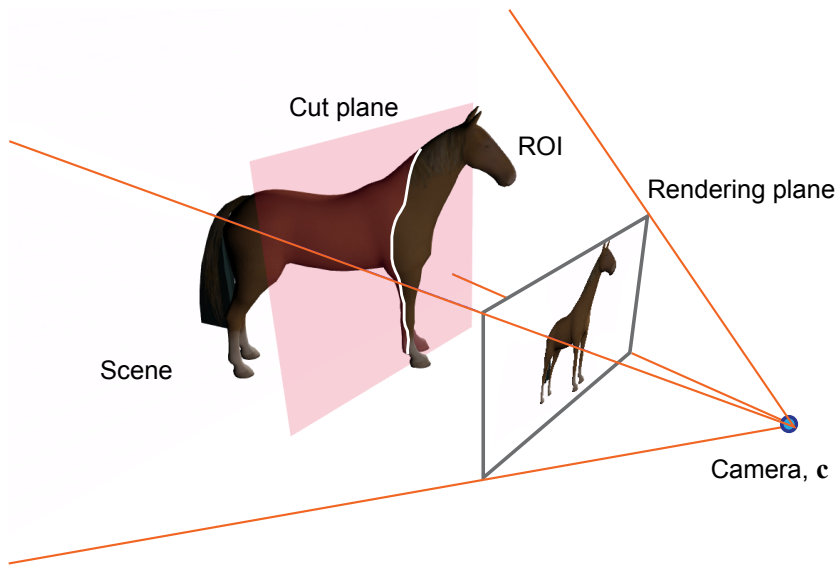


Figure 4.3 – The overall setup of the scene. We define a camera position and a rendering plane. For a given object in the scene, the user selects, through a cut plane, the object part (ROI) to be in 3D.

## 4.2 Approach

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 that provides 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  $V = \{\mathbf{v}_1, \dots, \mathbf{v}_n\} \subset R^3$  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. 4.3.

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. We generate the textures and UV maps to obtain the final result. In the following, we detail each step.

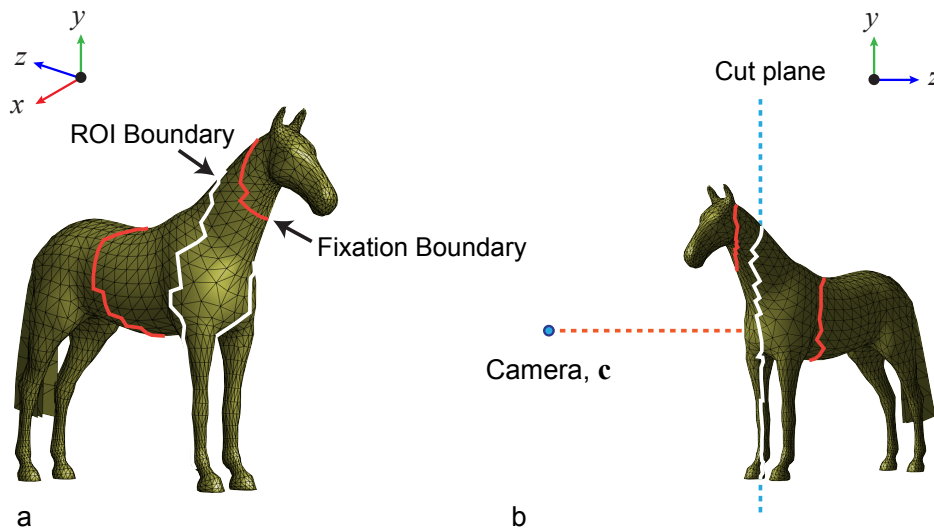


Figure 4.4 – The ROI boundary and fixation boundaries for the sample scene.

### 4.2.1 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 that aligns 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, which ensures the perception of a smooth transition both when viewed along the camera direction and along other directions.

We define all faces of the scene that lie 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. As our camera lies on the  $z$  axis of the world coordinate system, the  $z$  position of the cut plane  $h_z$  is considered to be the parameter governing the transition

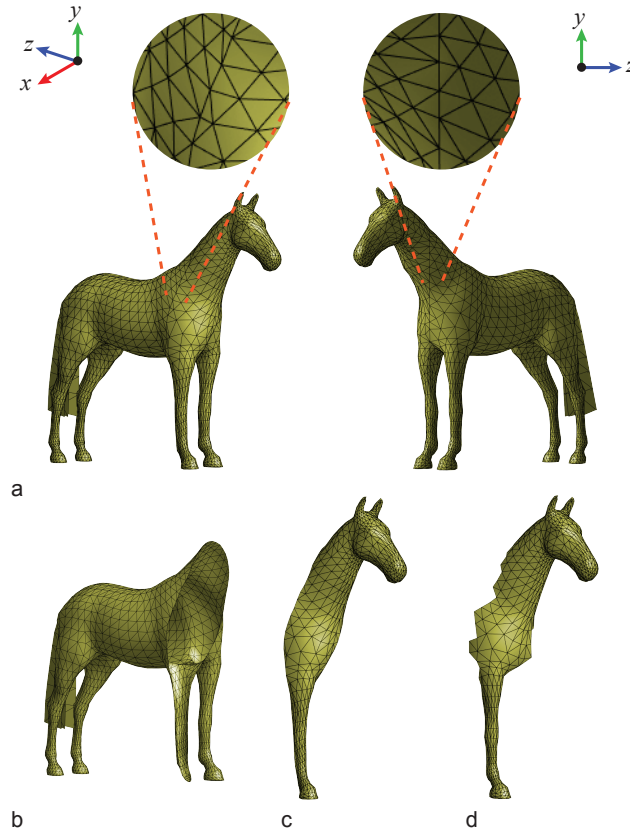


Figure 4.5 – a) The ROI boundary after the refinement operation. b) The non-ROI part after the scene object is cut. c) The ROI part after the scene object is cut along the boundary. d) The ROI without boundary refinement.

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:

$$v_i(z) \leq h_z + \beta \cdot \eta \quad (4.1)$$

where  $\beta \cdot \eta$  is the tolerance parameter,  $\eta$  is the average edge length in the scene and  $\beta$  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  $B = (v_1^b, \dots, v_m^b)$  of the ROI, where  $B \subset V$  and  $m$  is the number of boundary vertices. Boundary 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  $E^b \subset E$ , 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  $E^b$ , we then convert the edge set to an ordered

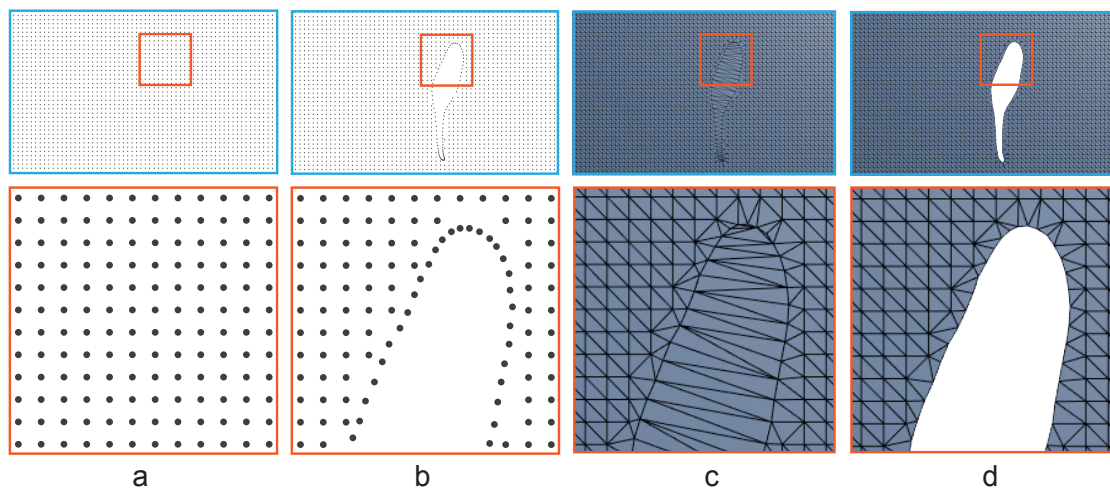


Figure 4.6 – a) The rendering grid points. b) Boundary vertices are added and the vertices inside the boundary polygon are removed. c) The initial geometry is created. d) Triangles inside the boundary polygon are removed.

polygon loop (Fig. 4.4).

As it can be seen in Fig. 4.4b, the ROI boundary is not perfectly aligned with the cut plane, due to the structure of the original geometry. Leaving the ROI like this 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, hence 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  $\Delta v_i$  for each vertex  $v_i$  as described by Sorkine et al. [82], see Appendix III.

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. 4.4. 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:

$$\begin{aligned}
 E(v_1, \dots, v_n) = & \sum_{i=1}^n \|\Delta v'_i - \Delta v_i\|^2 + \sum_{j=1}^m \|v_j^b - v_j^b\|^2 \\
 & + \sum_{k=1}^d \|v_k^f - v_k^f\|^2
 \end{aligned} \tag{4.2}$$

The first term in the optimization minimizes the square differences between original differential coordinates  $\Delta v_i$  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  $v^b$  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. 4.5a-c.

#### 4.2.2 Preparation of the Rendering Plane

We create a grid to represent the rendering plane, and we 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.

$$\begin{aligned}
 a &= \frac{\sigma}{m} \sum_{i=1}^m \|b'_i - b'_{i+1}\| \\
 q &= w/a \\
 r &= h/a
 \end{aligned} \tag{4.3}$$

where  $\sigma$  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. 4.6a. 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. 4.6b). 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 [85] over the grid points to obtain a mesh surface (Fig. 4.6c). 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 that is ready for attachment with the ROI (Fig. 4.6d).



### 4.2.3 Attaching the ROI

In order to obtain a final fully connected surface, we need to attach the ROI to the rendering grid. First we apply a 3D transformation to 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 the ROI boundary vertices on the rendering grid, we merge the vertices and the 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. 4.7).

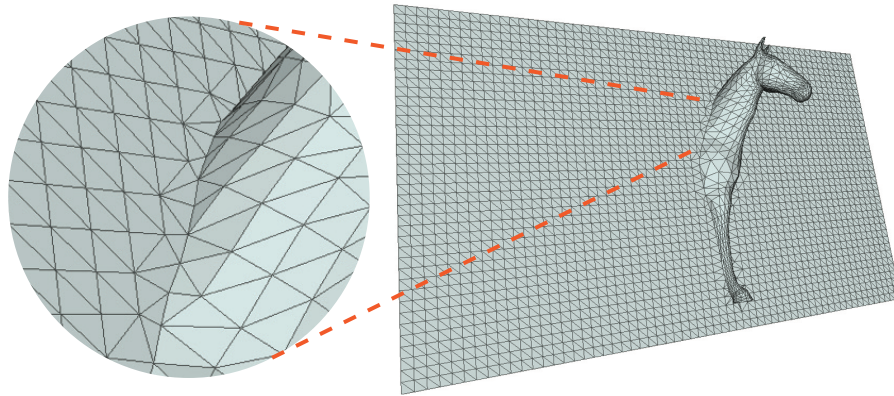


Figure 4.7 – Final geometry after the attachment operation.

### 4.2.4 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, similarly to the method of Weyrich et al. [36], 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:

$$\|(\frac{\partial v_i^d}{\partial x}, \frac{\partial v_i^d}{\partial y})\| > s \quad (4.4)$$

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

Next we assign depth values  $v_i^d$  to the flat part of the attached geometry as their  $z$  coordinates. We find differential coordinates of the new attached geometry as discussed in Section 4.2.1. 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.

$$E(v_1(z), \dots, v_n(z)) = \sum_{i=1}^n \|\Delta v_i^d(z) - \Delta v_i(z)\|^2 + \sum_{j=1}^h \|v_j^s(z) - v_j^s\|^2 \quad (4.5)$$

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. 4.8). In contrast with previous approaches [36], there is no need to apply scaling on the gradients for a smooth transition between the 2D and the 3D parts.

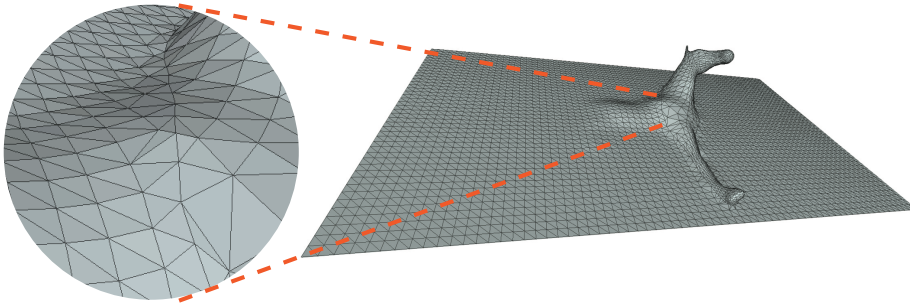


Figure 4.8 – Addition of bas-relief profiles.

#### 4.2.5 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. 4.3). After replacing the scene with the non-ROI (Fig. 4.5b), 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.

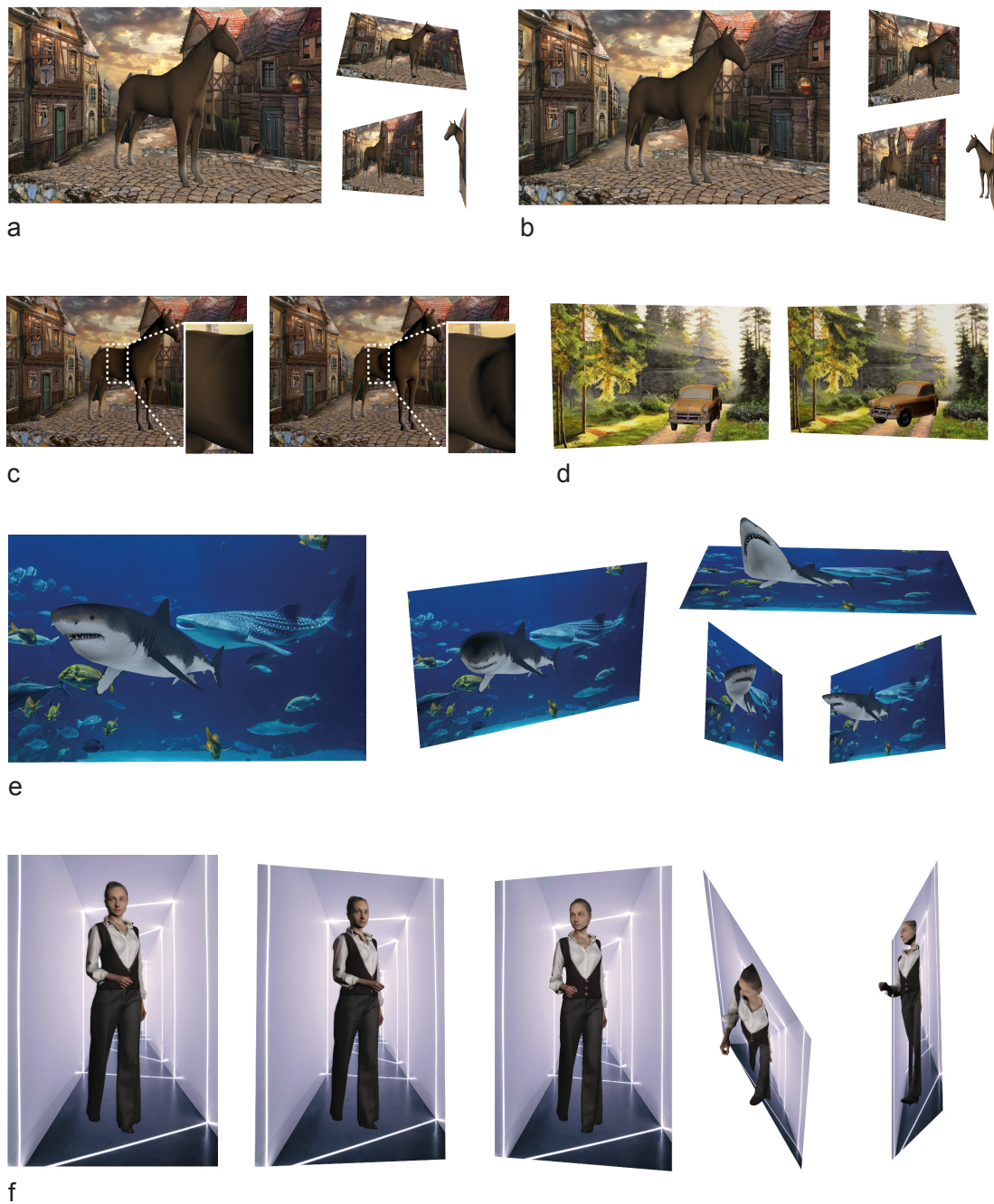


Figure 4.9 – a-b) Sculpture paintings generated with the horse model according to different cut positions. c) Strong directional light can create artefacts at the boundary of 2D and 3D scene elements, but the addition of bas-relief profiles decreases these artefacts to a large extent (left). d) Sculpture painting generated with the car model. e) Sculpture painting generated with the shark model. f) Sculpture painting generated with the woman model. (Best viewed on screen.)

For the 3D part, we use the original texture map and UV coordinates. As 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 [86, Chapter 7]. 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 use only ambient light with the newly baked texture. For the 3D part, the baked texture is directly used.

### 4.3 Results and Discussion

As shown in Fig. 4.9, 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 then 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, as the boundary refinement causes less deformations. We also create a planar grid that has 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.

#### 4.3.1 Lighting Artefacts

The geometry differences between flat and 3D parts can create lighting artefacts. Lighting artefacts occur 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 challenges 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 results 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 4.2.2. 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. 4.9c). The resulting sculpture paintings look good under many lighting conditions and the eye can be fooled from many viewing directions.

### 4.3.2 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 viewing independence depends on the content. The scenes including elastic and self-moveable content such as a horse, a deer, or a shark have view-independence larger 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 had changed its pose. However, as this kind of self movement is not possible for solid objects, they might be perceived as broken or deformed (Fig. 4.9f, 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.



Figure 4.10 – Sculpture paintings manufactured with a 3D printer.

### 4.3.3 Scene Design and Cut Position

Our camera position and rendering plane is predefined. Users can modify the input scene by changing its position, its scaling factor, and rotation angle. We also enable 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. 4.9a-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 with multiple-boundary polygons. The position of the cut plane provides an artistic control, as each cut position creates a slightly different composition.

### 4.3.4 Style Transfer

In addition to the main scene object, other objects in the scene provide creating a supporting environment. We provide an interactive environment, enabling designers to control their scene. Because of performance reasons, instead of creating a whole environment, background images can be used as well. In order to match the style of the background image and the textures of the scene object, we 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 [84].

### 4.3.5 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 inkjet printing on paper. Yet, we can generate nice results with the current technology (Fig. 4.10).

## 4.4 Conclusion

We have proposed a method to create visually convincing sculpture paintings from given 3D synthetic scenes. We enable the user to choose the parts to be represented 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 from 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

freedom to artists to design the overall input scene and to select the region of interest. Our method can be used in architecture, decoration, and households. Due 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 that incorporate sculpture paintings, can be easily conceived and produced.

#### **Acknowledgements**

The original 3D scene models used in the present chapter are obtained from TURBOSQUID, with the royalty free licences, <https://www.turbosquid.com/>





## 5 Synthesis in the Temporal Domain: Tempocodes

In this chapter, we present a method for hiding images in synthetic videos and reveal them by temporal averaging. The main challenge is developing a visual-masking method that hides the input image, both spatially and temporally. Our masking approach consists of temporal and spatial pixel-by-pixel variations of the frequency band coefficients that represent the image to be hidden. These variations ensure that the target image remains invisible both in the spatial and the temporal domains. In addition, by applying a temporal masking function derived from a dither matrix, we enable the video to carry a visible message that is different from the hidden image. The image hidden in the video can be revealed by software averaging, or with a camera by long-exposure photography. Our work can be applied in the secure transmission of digital information.



Figure 5.1 – A tempocode is a video (left) containing a hidden image that can be revealed through long-exposure photography of the video (right).

### 5.1 Introduction

We can hide an image in a video stream under the constraint that the temporal average of the video reveals the image. Specifically, the input image should remain invisible in each frame of

the video and should not become visible due to the temporal integration of consecutive frames by the human visual system (HVS). In order to achieve this, a visual-masking method that acts both in the spatial and in the temporal domain is required. Spatial masking inhibits orientation and frequency channels of the HVS. For temporal masking, any information coming from the target image by temporal averaging should be masked. As surveyed by Chandler [45], many methods of masking were developed for the spatial domain. However, masking in the temporal domain has not received much attention.

Our algorithm hides an input image within a video. The image is revealed by averaging, which is either achieved by pixelwise mathematical averaging of the video frames or by long-exposure photography (Fig. 1). We call the video hiding the input image *tempocode*.

In order to create such tempocodes, we apply the following self-masking model. We first decrease the dynamic range of the input image and decompose it into a certain number of frequency bands [87]. For each frequency band of the contrast-reduced input image, we generate temporal samples by sampling a selected masking function whose integration along a certain time interval gives the corresponding frequency band. We then reconstruct each video frame from the temporal samples derived from the frequency bands. We consider the following masking functions: random function, sinusoidal composite wave function, and a temporally-varying dither function. Using these functions, we generate different masking effects such as smoothly evolving videos and videos with visible moving patterns.

**Contribution.** By exploiting the limitations of the human visual system with respect to the temporal domain, we design an algorithm for creating special video seals. Such a synthesized video either appears as spatial noise or carries a visible message different from the hidden one. If the correct exposure time is set, the hidden image is revealed by a camera. Hiding an image in a video can be used as a security component within a framework for transmitting and authenticating electronic documents, e.g., electronic tickets or digital identity cards.

## 5.2 Self Masking

We now describe our approach for hiding an image in a video. The hidden information is not perceivable by the human eye, but the pixelwise average of the video over some time interval reveals the hidden image. With the correct exposure time, conventional and digital cameras can detect the hidden information. Software averaging over the video frames also reveals the image.

The main challenge resides in masking the input image with spatio-temporal signals that are a function of the input image. To achieve this, we present a visual-masking process that enables hiding the input image, in respect to both spatial and temporal perception.

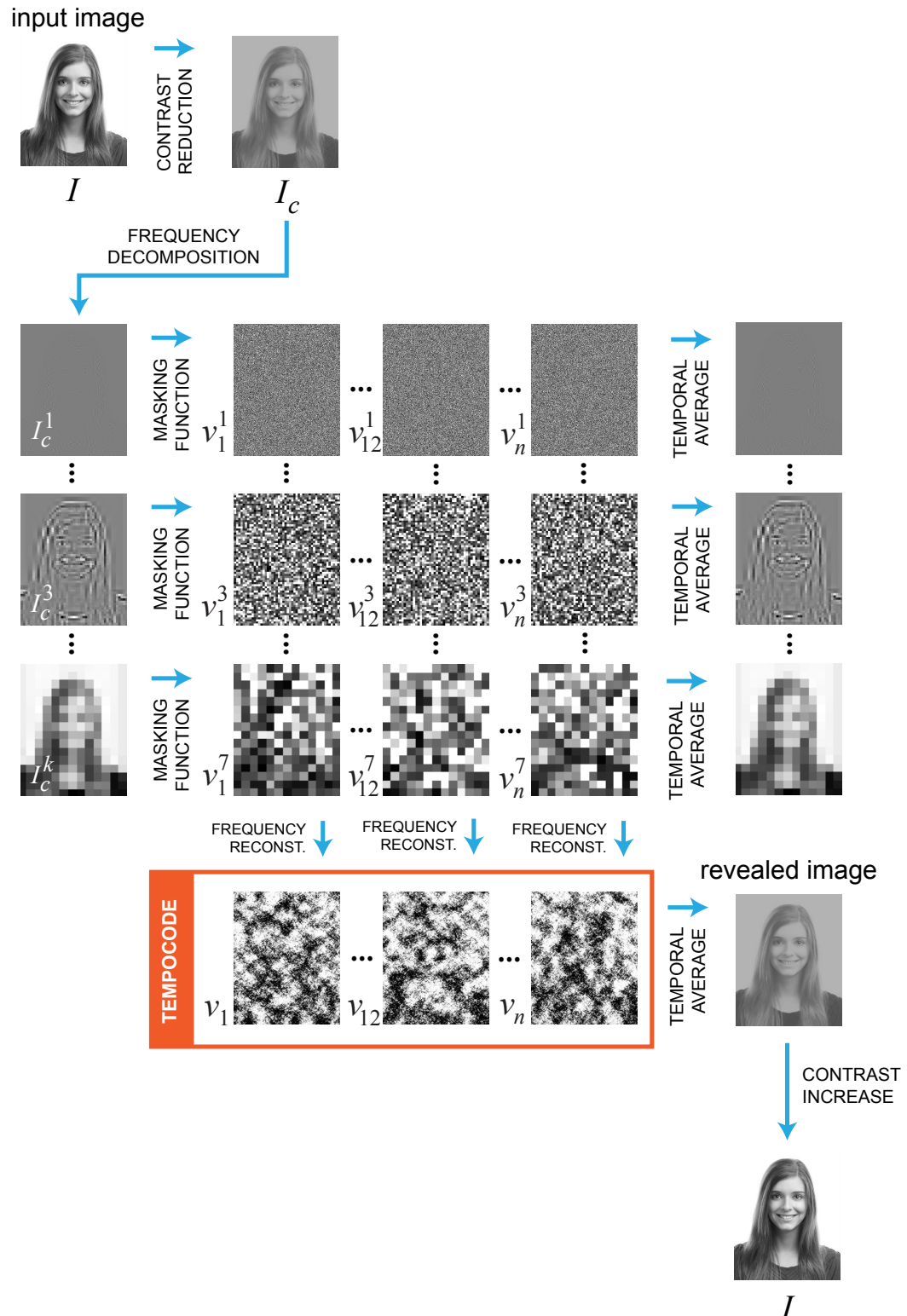


Figure 5.2 – An overview of our model. A tempocode is generated for an input image  $I$ . The resulting video has  $n = 24$  frames and is constructed with  $k = 7$  frequency bands. In the figure, only 3 frames and 3 frequency bands are shown.

## Chapter 5. Synthesis in the Temporal Domain: Tempocodes

---

In conventional visual-masking methods, masks and targets are different stimuli. However, in our method, the mask is defined as a function of the target image. We call this approach *self masking*.

We initially define the problem in the continuous domain. A constant target signal  $p$  is reproduced by the integration of  $f(t)$ , a time-dependent masking function, over a duration  $\tau$ :

$$p = \frac{1}{\tau} \int_0^{\tau} f(t + \delta) dt \quad (5.1)$$

In order to create spatial noise, a phase-shift parameter  $\delta$  is selected randomly at each spatial position. We assume that the display is linear. The target signal  $p$ , the duration  $\tau$ , and the phase shift  $\delta$  are predetermined parameters. The challenge resides in finding a function  $f(t + \delta)$  satisfying this integration and ensuring that the target signal is masked at each time point. In section 5.4, we discuss the different alternatives for the masking function  $f(t + \delta)$ .

In practice, our signals are not continuous as the target is a digital image and the mask is a digital video designed for modern displays. Let  $I(x, y)$  be a target image ( $0 \leq I(x, y) \leq 1$ ) to be masked into a video  $V$  having  $n$  frames. Initially, we reduce the contrast of the input image  $I$  by linear scaling. As discussed in Section 5.3, this is required in order to reach the *masking threshold*.

A multi-band masking approach is required to mask both high frequency and low frequency components. Applying the masking function solely on input pixels would mask only the high-frequency component. Therefore, we decompose the low-contrast target image  $I_c$  into spatial frequency bands. A Gaussian pyramid is computed from the target image  $I_c$ . To obtain the frequency bands, we compute the differences of every two neighbouring pyramid levels. In practice, we use a standart Laplacian pyramid with a 1-octave spacing between frequency bands [88]. Finally, for each contrast reduced pixel value  $I_c^l(x, y)$  in each band  $l$ , we solve a discretized instance of Eq. (5.1). Let  $t_1, \dots, t_n$  be a set of  $n$  uniformly spaced time points. Then the integral in Eq. (5.1) is approximated as follows

$$I_c^l(x, y) = \frac{1}{n} \sum_{i=1}^n v_i^l(x, y) \quad (5.2)$$

$$v_i^l(x, y) = f(t_i + \delta_l(x, y)) \quad (5.3)$$

where  $v_i^l$  is a frequency band  $l$  of the frame  $v_i$  at time point  $t_i$  of the resulting video. A different phase shift value  $\delta$  is assigned to each pixel  $(x, y)$  in each band  $l$ .

Once all bands  $v_i^l$  of each frame  $v_i$  are constructed, we sum the corresponding bands to obtain the final frame  $v_i$  at time point  $t_i$ :

$$v_i(x, y) = \sum_{l=1}^k v_i^l(x, y) \quad (5.4)$$

where  $k$  is the number of bands. An overview of the model is given in Fig. 5.2.

### 5.3 Contrast Reduction for Masking Purposes

As presented in Section 2, a masking signal with a certain contrast can mask a target signal that has a contrast smaller than the *masking threshold*. In our model, we always generate our mask with 100 percent contrast in order to enable a maximum contrast of the target image. We first reduce the contrast of the target image  $I$  and move the contrast reduced image to the center of the available range:

$$I_c(x, y) = \alpha \cdot I(x, y) + \frac{1}{2} - \frac{\alpha}{2} \quad (5.5)$$

where  $\alpha$  is the reduction factor and  $0 < \alpha < 1$ .

The amount of contrast reduction  $\alpha$  depends on the contrast, spatial frequency, and orientation of the target (see Sec. 2.2).

It is very important to select the correct contrast-reduction factor  $\alpha$  to reach the *masking threshold*. However, the input image consists of a mixture of locally varying contrasts, spatial frequencies, and orientations that affect masking. The contrast-reduction factor  $\alpha$  should be selected by considering the local image element that requires the largest amount of contrast reduction. Once this image element is masked, all other image elements are masked as well.

As our mask always has the maximum contrast, its contrast is not a variable in our model.

### 5.4 Masking Functions

Many different types of masking functions  $f(t + \delta)$  fulfill the requirements of Eq. 5.1. We can define a random function with uniform probability, a Gaussian function, a Bezier curve, a logarithmic function, or periodic functions such as a square wave, a triangle wave, or a sine wave. However, the following constraints need to be satisfied:

- Eq. (5.1) must have a solution for the selected function within the dynamic range of each frequency band.
- Masking must be achieved during the whole video  $V$ . In other words, any visual element that could reveal the target image  $I$  must remain invisible to the human eye.

In the following, we describe random, periodic, and dither masking functions.

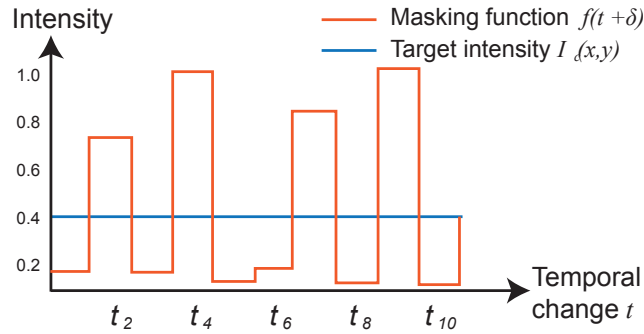


Figure 5.3 – Example of a discontinuous random function  $f(t)$  to mask the target image.

#### 5.4.1 Random Masking Function

Our random-masking function is made of  $n$  random uniformly distributed samples varying temporally for each pixel of each band. The mean of this uniform distribution is given by the intensity  $I_c^l(x, y)$  of the corresponding pixel of band  $l$ . Eq. (5.2) holds with an error that depends on the number of samples. If the number of samples is small, the error becomes larger. To enforce Eq. (5.2), we redistribute the error over all samples. Besides, the samples whose values are out of the allowed range are clipped. The remainders are redistributed equally to the other samples. This process is repeated until all samples are within the allowed range.

As shown in Fig. 5.6b, if the contrast of the target image is sufficiently reduced, the random function masks, to a large extent, the target image. However, this is only true when each frame is observed separately. When all frames are played as a video (e.g., at  $30\text{ fps}$ ), the target image is slightly revealed. This is due to the fact that the target image is well masked spatially but not temporally. The human visual system has a temporal integration interval of  $40 \pm 10\text{ ms}$  [53]. Therefore, a few consecutive frames can be averaged by the HVS. If we look at the signal in Fig. 5.3, the average of any two consecutive frames has a value close to the target intensity. A low-frequency masking function is therefore required to ensure temporal masking.

#### 5.4.2 A Sinusoidal Composite Wave

As we have seen in the previous section, a temporally continuous low-frequency masking signal is required to avoid revealing the target signal by temporal integration of the HVS. We

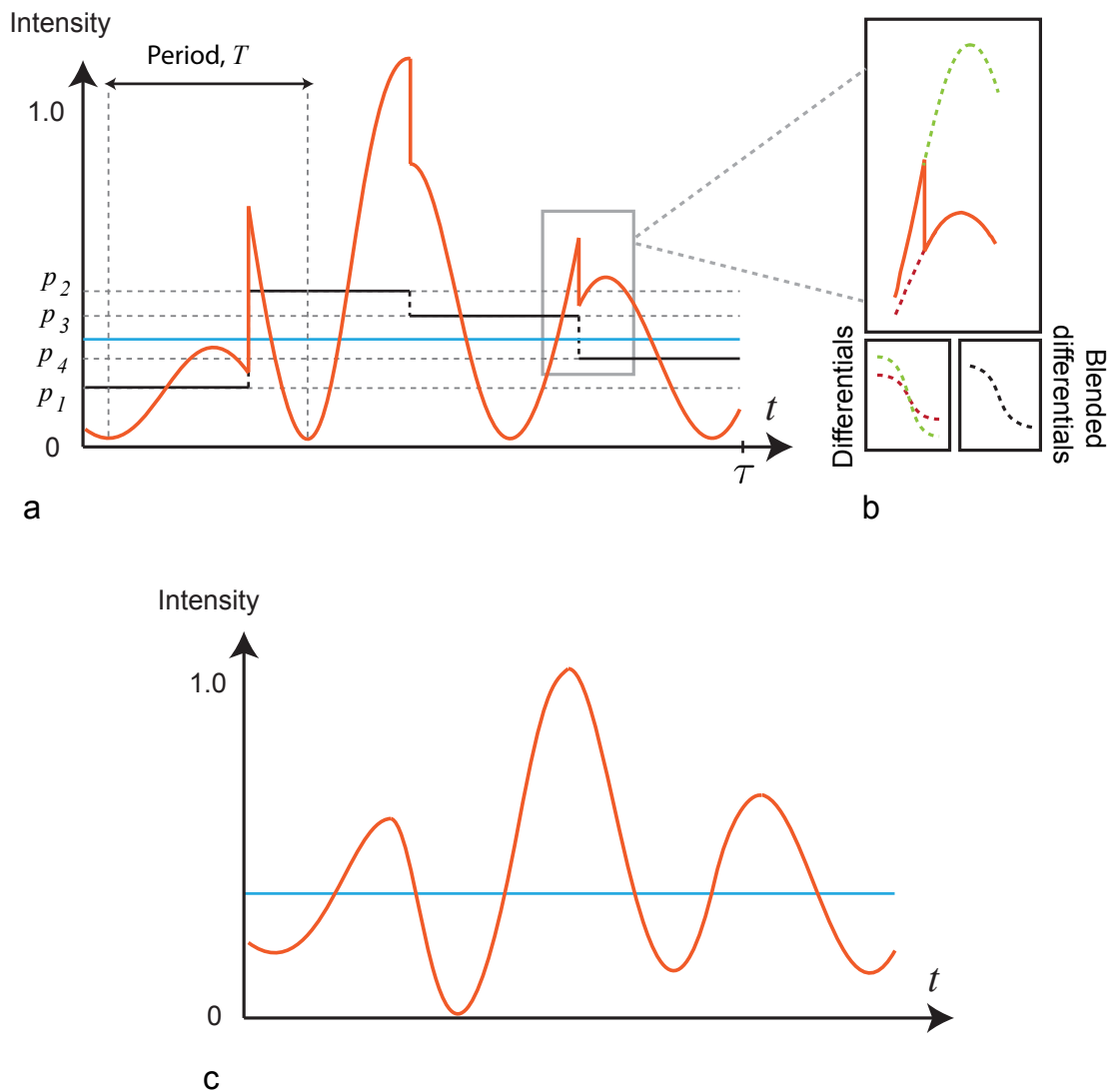


Figure 5.4 – The integration of a modulated wave yields the given target intensity  $I_c^l(x, y)$  (blue line). a) First 4 parent-samples  $p_1, p_2, p_3, p_4$  are generated. Their average gives the target intensity. Then for each parent sample, a simple sinusoid is generated by ensuring that its integration yields the parent sample. b) A refinement process over the modulated wave is applied to remove the discontinuities. c) After the optimization process, we obtain a smooth modulated wave.

thus propose a periodic function that results in spatial discontinuity and temporal continuity of the resulting video.

We use a sine function as our periodic function. Spatial juxtaposition of phase-shifted sine functions can reveal local parts of the target image. Therefore, instead of using a regular sine function, we create a sinusoidal composite wave by varying the function in amplitude for a

given number of segments.

In order to create  $m$  sine segments varying in amplitude, we first generate  $m$  uniformly distributed random parent-samples  $p_j$  for each pixel of each band, ensuring that their mean is  $I_c^l(x, y)$ :

$$I_c^l(x, y) = \frac{1}{m} \sum_{j=1}^m p_j \quad (5.6)$$

Since we have a small number  $m$  of parent-samples (e.g.,  $m = 4$ ), the mean  $I_c^l(x, y)$  will not hold. Therefore, we redistribute the error across the samples. Next, for each parent-sample  $p_j$ , we establish a function  $f_j(t + \delta)$  in the form of Eq. (5.1) such that

$$p_j = \frac{1}{\tau_e^j - \tau_s^j} \int_{\tau_s^j}^{\tau_e^j} f_j(t + \delta) dt \quad (5.7)$$

where  $\tau_s^j = (j - 1) \times \frac{\tau}{m}$  is the start time,  $\tau_e^j = (j) \times \frac{\tau}{m}$  is the end time,  $j \in [1, \dots, m]$  is the index of each parent-sample, and  $\tau$  is the total duration of the video to be averaged.

We define the masking function  $f_j(t + \delta)$  for each parent sample as a continuous section of a sine in a form that is analytically integrable and that lies within the dynamic range for most of its values.

$$f_j(t + \delta) = k_j \cdot \sin\left(2\pi \frac{t}{T} + \delta\right) + k_j \quad (5.8)$$

where  $k_j$  is the amplitude,  $\delta$  is the phase shift ( $0 \leq \delta \leq 2\pi$ ) and  $T$  is the period. As shown in Fig. 5.4a, the period  $T$  and the duration of video  $\tau$  have different values. The total duration  $\tau$  of video is given by the user.

By inserting Eq. (5.8) into Eq. (5.7), we can express  $k_j$  in function of the other parameters:

$$k_j = \frac{p_j(\tau_s^j - \tau_e^j)}{\tau_s^j - \tau_e^j + \frac{T(\cos(\frac{2\pi\tau_e^j}{T} + \delta) - \cos(\frac{2\pi\tau_s^j}{T} + \delta))}{2\pi}} \quad (5.9)$$

For each pixel of each frequency band, these  $m$  functions  $f_j(t + \delta)$  of parent samples are sam-



pled with one sample per video frame (Fig 5.4a). The averages are enforced by redistributing the errors over the samples. According to Eq. (5.7), the average of each sinusoidal section gives the value of a parent sample. Thus, the average of all  $n$  samples gives the target intensity of the considered band  $I_c^l(x, y)$ .

In order to ensure a phase continuity between the sinusoidal segments, we select the phase shift  $\delta$  randomly only for the first sinusoidal segment  $f_j(t + \delta)$ . For all other functions associated with parent samples, we use the current phase  $\delta$  and the current period  $T$ . Nevertheless, due to the variations of the amplitudes, we obtain a non-continuous composite signal. These discontinuities appear at the junctions between successive sinusoidal segments (see Fig 5.4a) and are visible in the final output video.

To remove the discontinuities at the junction points, we apply a refinement process by using differential values. From the samples of the composite wave, we first calculate the differential values by taking the backward differences:  $\Delta v_i^l(x, y) = v_i^l(x, y) - v_{i-1}^l(x, y)$ . We then blend the differential values of the end part of a sinusoidal segment with those at the starting part of the following sinusoidal segment (see Fig. 5.4b).

With the blended differential values, we re-calculate the intensity values for each pixel of each band by minimizing the following optimization function:

$$\begin{aligned}
 E(v_1^l, \dots, v_n^l) = & \sum_{i=1}^n \|\Delta v_i^l(x, y)' - \Delta v_i^l(x, y)\|^2 \\
 & + \|I_c^l(x, y) - \frac{1}{n} \sum_{i=1}^n v_i^l(x, y)'\|^2 \\
 & + \sum_{b=1}^m \|v_b^l(x, y)' - v_b^l(x, y)\|^2
 \end{aligned} \tag{5.10}$$

where  $n$  is the total number of frames. The first term in the optimization minimizes the square differences between blended differential values  $\Delta v_i^l(x, y)'$  and the differential values  $\Delta v_i^l(x, y)'$  of the new intensities in the solution set. The second term is a constraint to guarantee that the overall average  $I_c^l(x, y)$  of the new intensities  $v_i^l(x, y)'$  is still satisfied. The third term preserves the overall shape of the signal, by fixing the center sample of each sinusoidal segment as a constraint.  $b$  represents the index of center sample for each parent sample.

This optimization is solved as a sparse linear system. We obtain a smooth signal (Fig. 5.4c). The deviations from the average  $I_c^l(x, y)$  caused by the optimization are redistributed over the samples.

As shown in Fig. 5.6c (bottom row, inset), the sinusoidal composite wave successfully masks the target image in both the spatial and temporal domains.

### 5.4.3 Temporal Dither Masking Function

A sinusoidal composite wave enables masking the target image both spatially and temporally. However, the visible part, the tempocode video, does not convey any visual meaning. We thus propose to replace the spatial noise with meaningful patterns. For this purpose, we make use of artistic dither matrices [89].

When printing with bilevel pixels, dithering is used to increase the number of apparent intensities or colors [90]. A full-tone color image can be created with spatially distributed surface coverages of cyan, magenta, yellow, and black inks. The HVS integrates the tiny inked and non-inked areas into the desired color.

A dither matrix includes threshold values indicating at which intensity levels pixels need to be inked. Artistic dithering enables ordering these threshold levels so that for most levels turned-on pixels depict a meaningful shape [89]. We adapt artistic dithering to provide visual meaning to tempocode videos.

We repeat the selected dither matrix, horizontally and vertically, to cover the whole frame. We then animate the dither matrices [91]. The animation can be achieved by a uniform displacement of the dither matrices at successive frames. As shown in Fig. 5.5, for a single pixel, the threshold values vary over time. At any time point in the video, the current dither threshold determines if the pixel is white or black. Accounting for the varying dither thresholds over time, we can determine a dither input intensity, thus ensuring that the average of the resulting dithered black and white pixels yields the target intensity (Eq. 5.1, see Fig. 5.5b). Instead of finding such a dither input intensity, we directly assign white or black to the successive temporal dither threshold levels. For a first step, we describe the algorithm without frequency-band decomposition as follows:

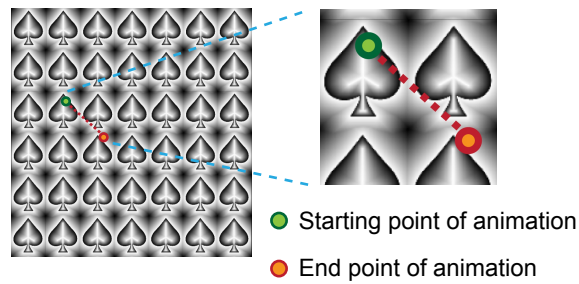
- Find the ratio  $r_{wb}$  of white to black pixels to obtain the target intensity  $I_c(x, y)$ . Then derive the number  $w$  of white pixels. This is calculated as follows:

$$r_{wb} = \frac{I_c(x, y)}{1 - I_c(x, y)} = \frac{w}{n - w} \quad (5.11)$$

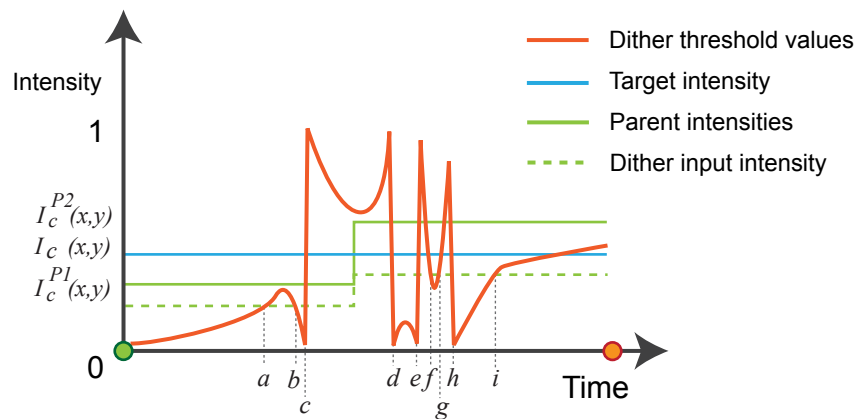
$$0 \leq I_c(x, y) \leq 1$$

where  $n$  is the total number of frames.

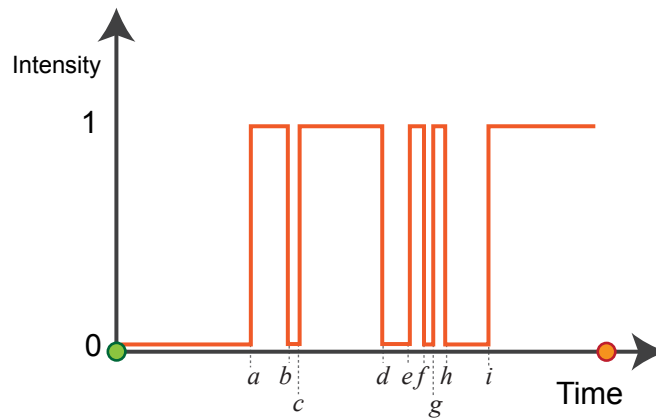
- For each spatial pixel, sort its dither threshold values that change temporally in descending order.
- Replace the dither values by 1 (white pixel) for the first  $w$  pixels in the sorted list and the rest by 0 (black pixel).
- Revert the new pixel values back to their original indices (e.g. frame number).



a) Animating dither profiles



b) Thresholds for a pixel over time



c) Final values for a pixel over time

Figure 5.5 – a) An artistic dither matrix representing a spade is repeated horizontally and vertically to cover the whole frame. b) For each frame of the output video, the array of dither matrices is translated in the direction of the desired animation. This creates, for each pixel, a succession of dither thresholds over time. An optimum dither input intensity is determined for the two parent samples separately. c) The pixels above the dither input intensity are assigned black and the others white. The overall average gives the target intensity.

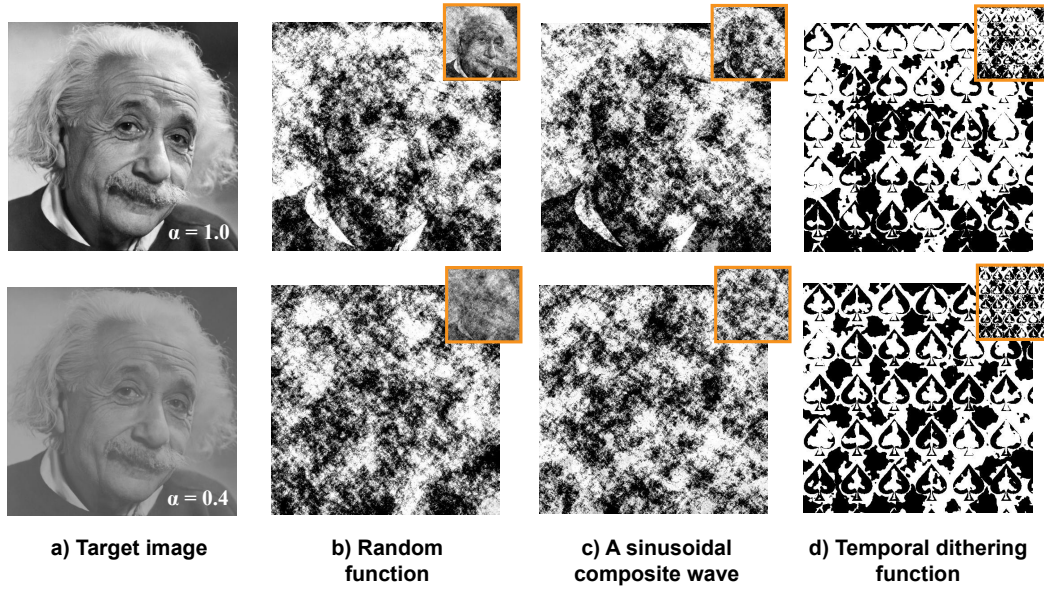


Figure 5.6 – Sample frames from the tempocodes generated with different masking functions with the following parameters: Duration  $\tau = 4s$ , frame rate =  $60fps$ , the period  $T = 1.65s$ , and the number of frequency bands  $k = 7$ . In the first row, the results are generated with a target image having no contrast reduction ( $\alpha = 1.0$ ). None of the functions can fully mask the target image. In the second row, the contrast of the target image is reduced ( $\alpha = 0.4$ ). For a single frame, all functions can mask the target image. However, when a few consecutive frames are averaged by the HVS temporal integration, the random function (b) reveals the target image. The two other methods (c and d) are able to hide the target image, not only spatially but also temporally. The insets on the right top corner of the frames show the average of 4 consecutive frames as a simulation of the HVS temporal integration.

For an exact 8-bit representation of the target image,  $n = 256$  frames are required. With a lower number of frames, one may use additional gray levels or have an approximation of the target image.

This algorithm satisfies one of our conditions, i.e., the average of the frames yields the target image (Eq. (2)). However, a multi-band decomposition cannot be carried out with the dithered binary images as they are bilevel. As shown previously, the multi-band decomposition was an important component for masking the target image. To overcome this problem, we create two parent frames  $I_c^{P1}$  and  $I_c^{P2}$  from the input image  $I_c$  by using the random masking function applied on each band of image  $I_c$ , as described in Sec. 5.4.1. For these two parent frames, due to the multi-band decomposition, the target is masked spatially (e.g., Fig. 5.6b). Then for each of the two parent frames, we create  $\frac{n}{2}$  frames by using the temporal dither function as described above. In respect to Fig. 5.2, this means that we apply the dithering operation to tempocode frames ( $v_1$  and  $v_2$  as the parent samples). The levels and corresponding pixel values are shown in Fig 5.5 for a single pixel. Finally we get  $n$  dithered frames forming our final video  $V$  in which the target is successfully masked, as shown in Fig. 5.6d (see Fig. A-X in the Appendix).



Figure 5.7 – Sample tempocode frames (1) generated with the different input images and the different dither matrices. The hidden images can be revealed by averaging (2). An inverse contrast-reduction operation yields the original input image (3). In all the cases, the target image is recovered by software averaging the original tempocode frames. We have the following parameters (a) for the woman,  $\alpha = 0.4$ , (b) for the lion  $\alpha = 0.5$ , (c) for the QR code  $\alpha = 0.2$ , and (d) for the text,  $\alpha = 0.3$ .

## 5.5 Averaging

Tempocodes can be revealed by averaging. A pixelwise average of the original tempocode video yields the input target image (Fig. 5.2). For the random function and the sinusoidal composite wave, with floating point numbers, contrast reduction of the input image and masking is performed without losing precision. For dithering, some details of the original image can get lost due to the binarization of the signal (e.g,  $n < 256$ ) and possibly due to low-pass filtering of the target image in order to obtain smooth tempocodes.

Instead of averaging the tempocode by software, a camera can be used to average with long-exposure photography and to reveal the target image. The duration of temporal averaging can be controlled with the shutter speed [92]. Furthermore, smartphones can be used for averaging. By placing markers around the video, the projective distortion is removed and the rectified frames are averaged.

### 5.6 Results

The methods for generating tempocodes are described for grayscale target images. For color images, we use exactly the same procedure, and apply it to each color channel separately. Reaching masking thresholds by reducing the intensity range of the target image is critical to achieve masking. Thus, for color target images, we reduce the intensity range by considering the luminance channel. The chromatic contrast sensitivity of the HVS is lower than the achromatic contrast sensitivity. Once the masking threshold is reached for the luminance channel, the thresholds are also reached for the chromatic channels.

In order to illustrate the dissimilarity between the original image and the computed tempocode results, we use an image-similarity metric. We calculate the multi-scale structural similarity index (MS-SSIM) defined by Wang et. al. [93] between the original image and each frame of the output. The MS-SSIM is calculated considering luminance differences, contrast differences, structural differences, and viewing conditions. The MS-SSIM yields a similarity index value between 0 and 1. A high MS-SSIM index value means that the given two images are very similar [93]. We evaluate the dissimilarity between the hidden target image and the tempocode frames. We expect MS-SSIM values to be very small when the contrast reduction  $\alpha$  is sufficient. We compare each frame of the tempocode with the input image and calculate a mean MS-SSIM index for the whole video by averaging the results of each frame. We define dissimilarity as 1 minus the mean MS-SSIM index. In Fig. 5.8, the dissimilarity between tempocode frames that rely on the sinusoidal composite-wave function and the hidden image is shown for three different target images.

For photographic images (i.e., Einstein and lion) the dissimilarity is very high, even for a low-contrast reduction. The reason is that the frequency bands within the image have values far from their extrema, i.e., it is easier to generate a masking function. However, for a QR code, this is not the case. Most frequency bands use the full-intensity range. This requires the contrast of the input image to be strongly reduced (i.e.,  $\alpha = 0.2$ ) in order to achieve masking.

Any target image can be masked by reducing its contrast. We let the user select  $\alpha$  heuristically. From our experience,  $\alpha = 0.5 \pm 0.1$  enables masking most types of photographic images. For images, including texts and structures with high contrast (e.g., binary images), contrast reduction needs to be as much as  $\alpha = 0.3 \pm 0.1$ . Fig. 5.7 shows several tempocodes produced with different input images that require different contrast reduction factors  $\alpha$ .

As our contrast reduction is linear, the original target image can be obtained by multiplying the revealed images by the inverse of the intensity-range reduction (i.e., the inverse of Eq. (5.5)).

Our method introduces a screen-camera channel for hiding information by simple averaging. This can inspire future works for different applications. The encoding is complex, but the decoding is very simple. Thus, hidden images can be revealed by non-expert users, but not created. The present method does not compete with existing watermarking or steganographic

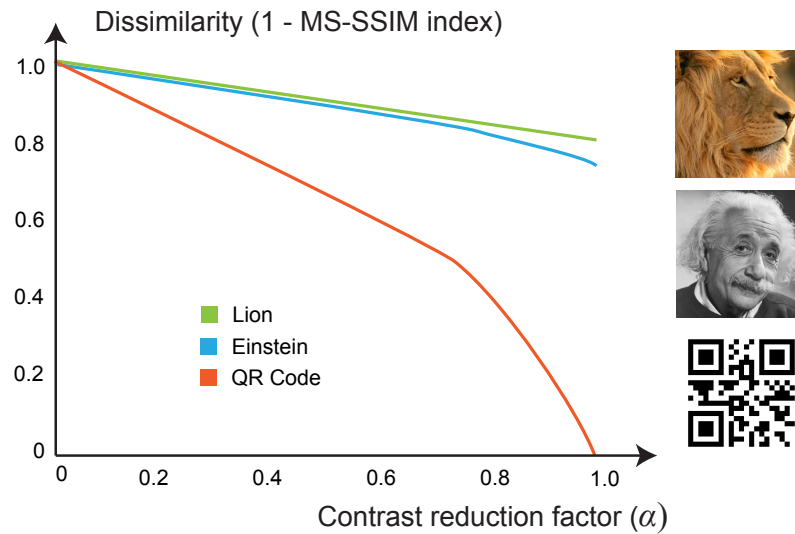


Figure 5.8 – The dissimilarity expressed as one minus the mean SSIM index for three different target images. Each frame of the output is compared with the target image. We compute the average dissimilarity over all frames. As we decrease  $\alpha$ , the dissimilarity increases differently for the target images.

methods that require complex decoding procedures. It can be rather used as a first level secure communication feature. More and more security applications, such as banking software use smartphones to identify codes that appear on a display. In the present case, instead of directly acquiring the image of a code, the smartphone might acquire a video that incorporates that code. For example, instead of showing a QR code on an electronic document directly, our method can be used to hide it. Hiding a message into a video can be seen as one building block within a larger security framework.

## 5.7 Conclusion

We have proposed to hide grayscale or color images in videos whose frames mask them, both spatially and temporally. The hidden image can be revealed by temporal pixel-wise averaging of the video. Many companies would like to provide their customers electronic tickets and identity cards. This necessitates new security features for protecting digital documents against forgery. Tempocodes have the potential to become a building block within a document-security framework. For authentication, smart phones can reveal the images or messages hidden into the tempocodes.





## 6 Conclusions

In this thesis, we have made a short introduction to artistic synthesis and have presented three different artistic-synthesis methods based on new computational frameworks. These methods can inspire the development of new computational artistic styles and help artists to create new types of artworks.

Our high-relief work is a first work that provides a comprehensive method to producing high reliefs from 3D synthetic objects. Although high relief is a common sculpting form, it has not had much attention from the computer graphics community. 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 synthesis. Instead, our method is based on differential coordinates in order to bring scene elements to the relief plane and to preserve 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 local compression. High-relief synthesis is controlled by user-defined parameters to adjust the depth range, as well as to adjust the placement of the scene elements with respect to the relief plane.

In sculpture paintings, we have presented 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 different relationships between 2D and 3D at each viewing angle. As the viewing angle is changed, we can clearly see 3D elements emerge 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 can select 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

## Chapter 6. Conclusions

---

the 3D parts of the object, we include bas-relief profiles in the neighbourhood of 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.

In the final work, we have presented a method for hiding images in synthetic videos and have revealed them by temporal averaging. The main challenge is to develop a visual-masking method that hides the input image, both spatially and temporally. Our masking approach consists of temporal and spatial pixel-by-pixel variations of the frequency band coefficients that represent the image to be hidden. These variations ensure that the target image remains invisible, both in the spatial and the temporal domains. In addition, by applying a temporal masking function derived from a dither matrix, we enable the video to carry a visible message that is different from the hidden image. The image hidden in the video can be revealed by software averaging, or with a camera, by long-exposure photography. Our work could find applications in the secure transmission of digital information.

Our works can inspire future works both in the artistic and in the computational domains. With the recent developments in the 3D printing, a great variety of the objects with different colors and materials can be manufactured rapidly, accurately, and cost-efficiently. This may motivate the artists to design the compositions with the help of the computer program and to print them in 3D. In this regard, our methods for high reliefs and sculpture paintings can be useful tools to explore new styles. Our tempocode method can inspire artists to create video art hiding a message, different than its applications in the security domain.

# A Appendix

## I. Notation

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

## 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  $g$  (Fig. A-II).

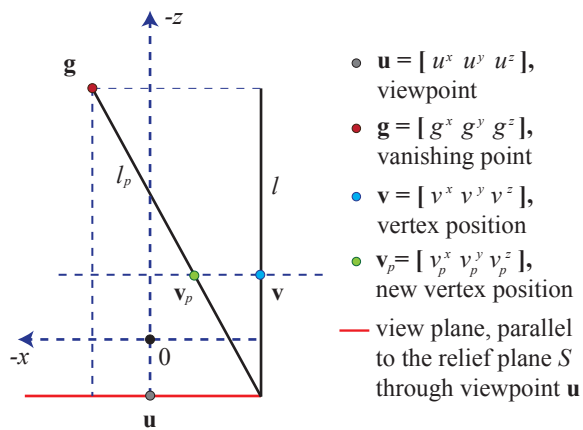


Figure A-II - One point perspective deformation.

A vertex  $\mathbf{v}$  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

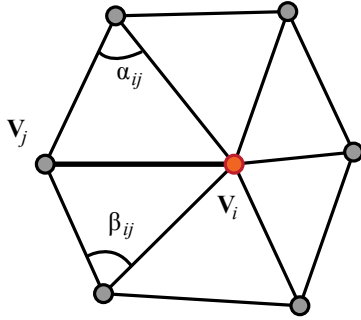


Figure A-III - 1-ring neighbours of a vertex.

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  $\mathbf{v}_i$ ,  $\omega_j$  is the cotangent weight for neighbour vertex  $\mathbf{v}_j$  (Fig. A-III) and  $|\psi|$  is the Voronoi area acting as normalization factor [82].

---

#### IV. New relief distances of attenuation points

In order to find differential values and minimize Eq 4.3, we proceed as recommended by Sorkine et al [76] 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  $\Delta a_i^z$  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  $\mathbf{D} = \text{diag}\{d_1, \dots, d_n\}$  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. 3.3, 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  $\mathbf{c}_j = \mathbf{a}_j$  indexed from  $b$  to  $k$  as indicated in Eq. 3.6, the system to solve by the least square method is the following:

$$\begin{pmatrix} \mathbf{L}_{k \times k} \\ \mathbf{0} \mid \mathbf{I}_{(k-b+1) \times (k-b+1)} \end{pmatrix} \mathbf{h}' = \begin{pmatrix} (\Delta\mathbf{h}') \\ \mathbf{c} \end{pmatrix}$$

#### 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  $\Delta V$ . 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  $\mathbf{D} = \text{diag}\{d_1, \dots, d_n\}$  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 [82].

$$\begin{pmatrix} \mathbf{L}(1 : m-1, 1 : n) \\ \mathbf{0} \mid \mathbf{I}_{(n-m+1) \times (n-m+1)} \end{pmatrix} \mathbf{V}'(1 : n, 3) = \begin{pmatrix} \Delta\mathbf{V}(1 : m-1, 3) \\ \mathbf{V}'(m : n, 3) \end{pmatrix}$$

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 second 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. 3.8). Then we recompute the values of these  $r$  vertices by adding the ones not having errors as positional constraints in the system:

$$\begin{pmatrix} \mathbf{L}'(1:r, 1:n) \\ \mathbf{0} \mid \mathbf{I}_{(n-r+1) \times (n-r+1)} \end{pmatrix} \mathbf{V}''(1:n, 3) = \begin{pmatrix} \Delta \mathbf{V}'(1:r, 3) \\ \mathbf{V}'(r+1:n, 3) \end{pmatrix}$$

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.

## VII. Supplementary Results for High Reliefs

In order to obtain the final relief geometry, we use two fitting steps, construction and restoration, instead of using one step fitting methods such as bi-Laplacian, higher order Laplacians, or other blending approaches such as biharmonic weights. The previous mesh deformation works mostly consider a certain region of interest (ROI), where the ROI is bounded by a stationary belt of anchors. On the other hand, our control points are sparsely distributed over the scene which results in dents or raises around control points when the positional constraints are fixed as hard constraints in the linear system (Fig. A-VII.1c). We observe that when the positional constraints are inserted into the system in the least square sense (Fig. A-VII.a) or higher order differential coordinates are used (Fig. A-VII.b), the errors around attenuation points are smoothed but they cause important scene deformations. We prefer to fix positional constraints and solve the system (Fig. A-VII.c), then re-construct the parts having high errors by using high order differentials (Fig. A-VII.d). As shown above, our two steps fitting procedure works better than the usage of higher order differentials or solving positional constraints in the least square sense in a single step.

Changing parameters enable creating different types of high reliefs as shown in Figures A-VII.2-9.

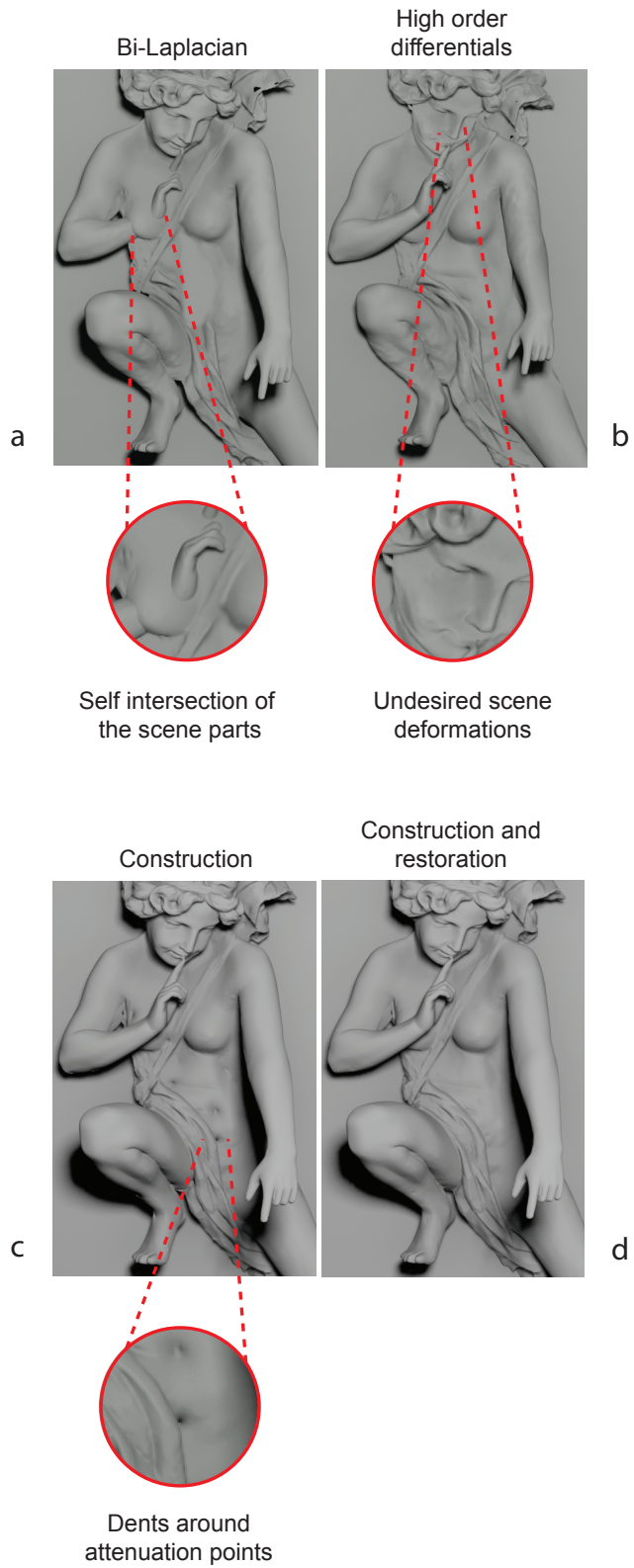


Figure A-VII.1 - Different construction methods for high reliefs.

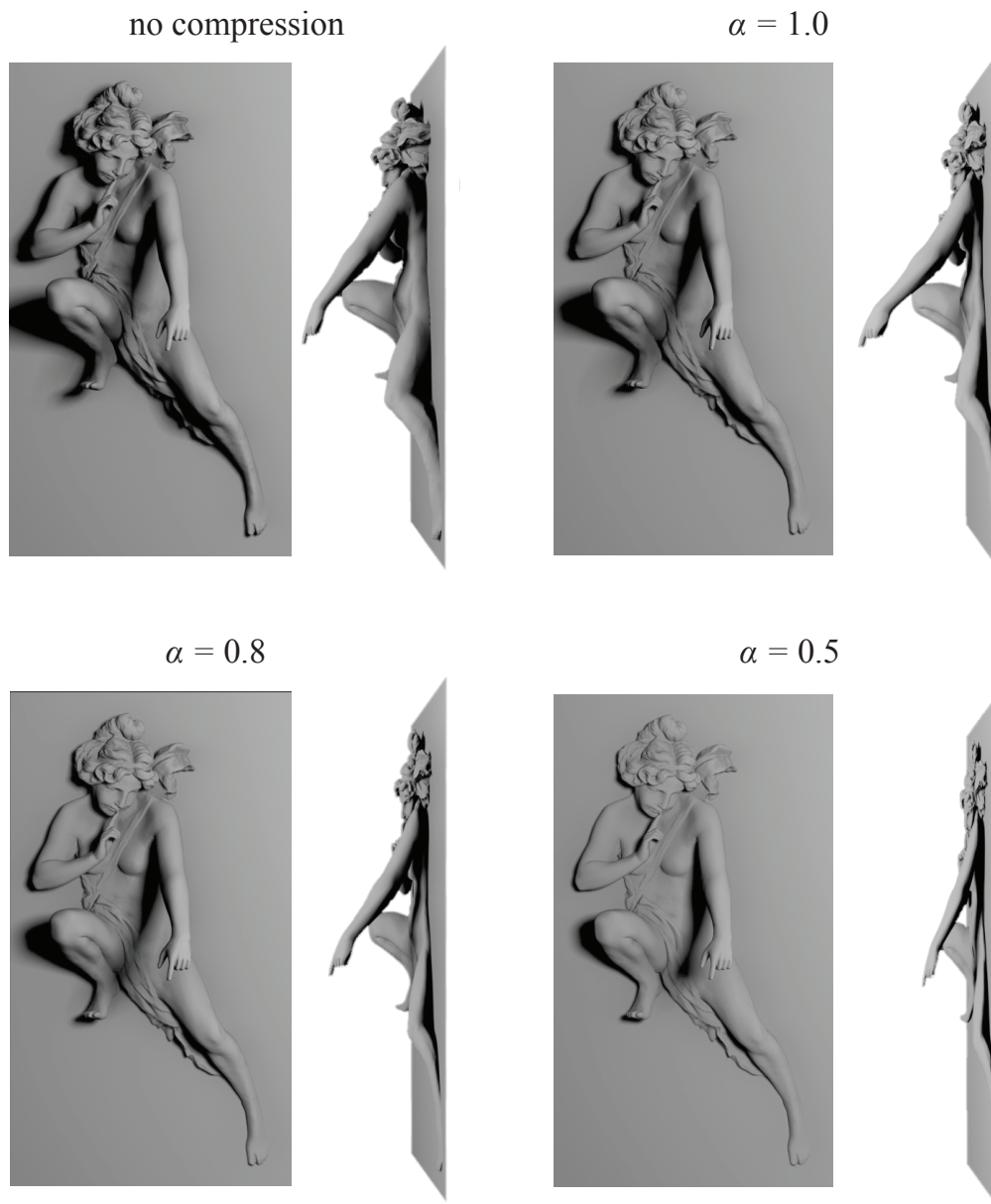


Figure A-VII.2 - Decreasing compression factor  $\alpha$  reduces the depth range but results in the loss of details. ( $k = 128$ ,  $\beta = 0.8$ )



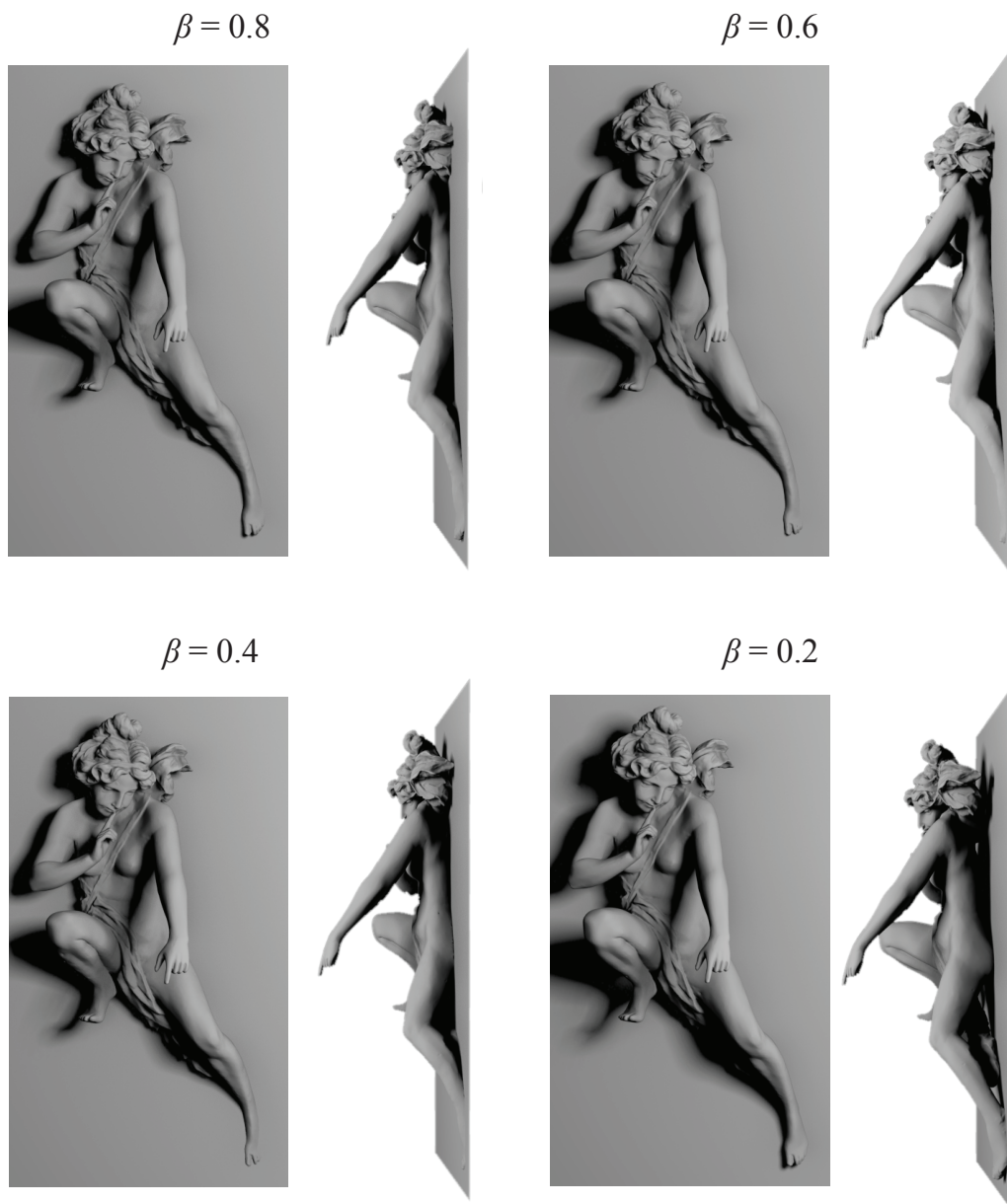


Figure A-VII.3 - Changing attenuation factor  $\beta$  determines how much the scene sinks in the relief plane ( $k = 128$ , no compression).

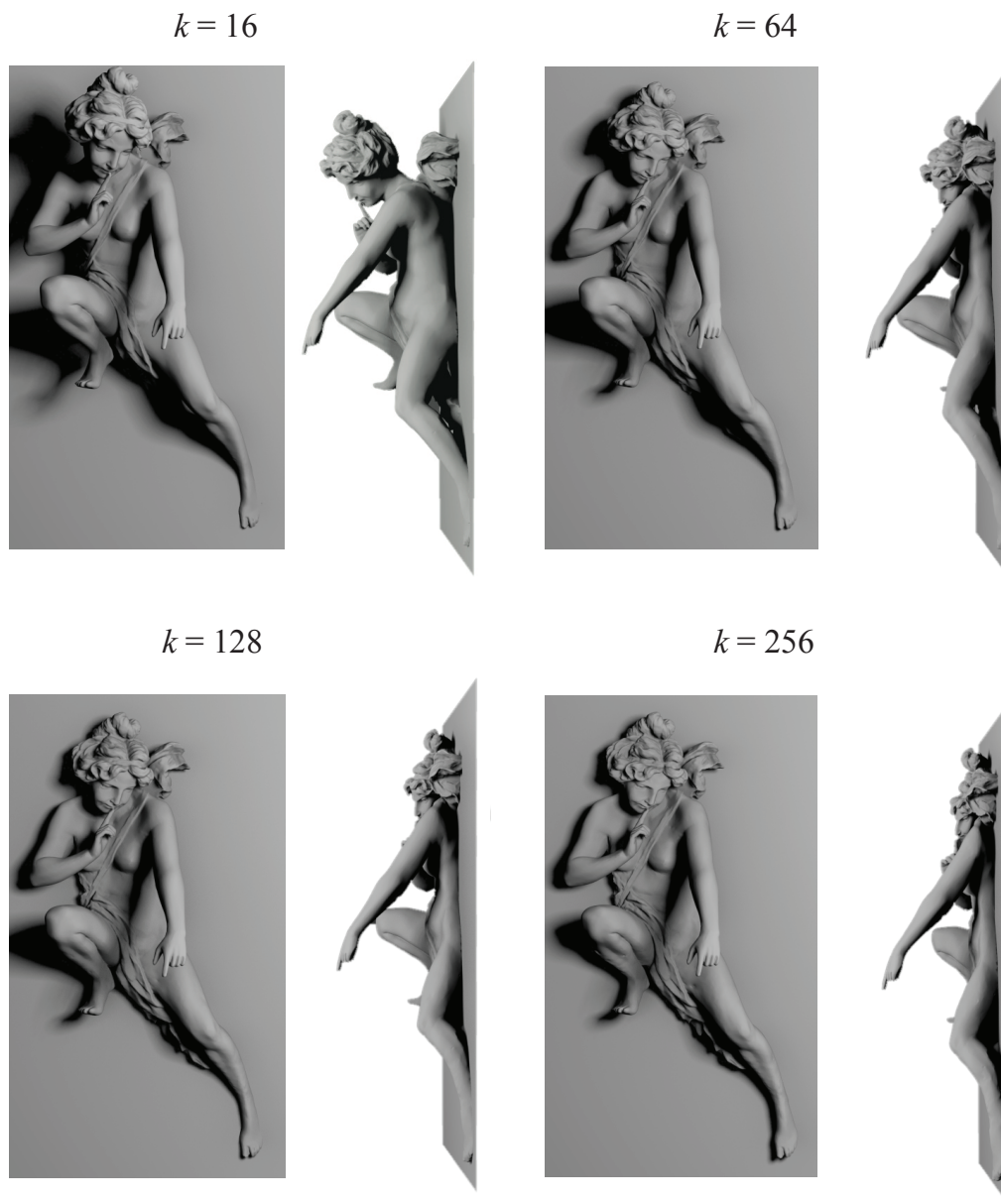


Figure A-VII.4 - Increasing the number  $k$  of the attenuation points increases the amount of scene elements that are brought towards the relief plane and adjusts the depth range ( $\beta = 0.8$ , no compression).

---

$k = 96, \alpha = 1.0$

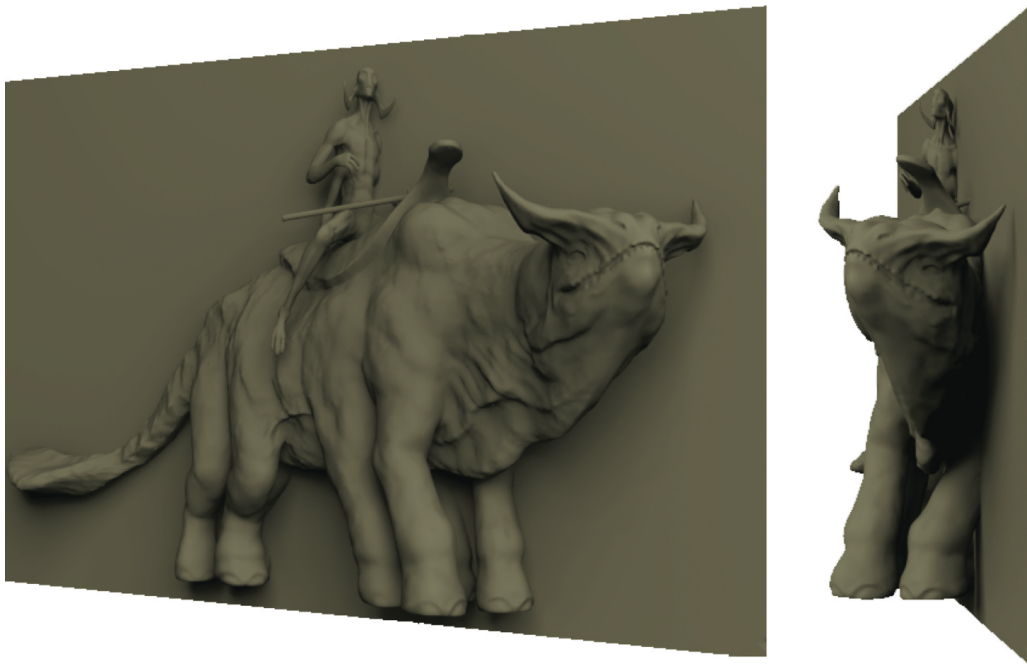
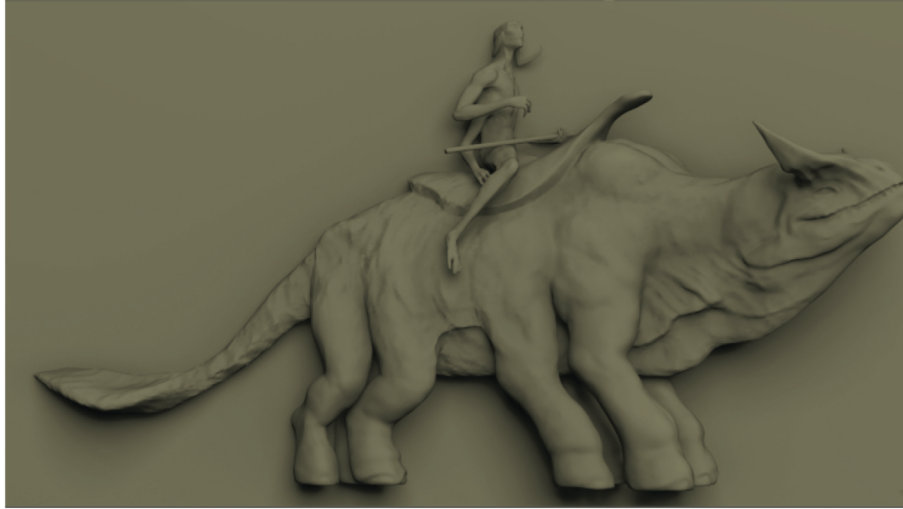


Figure A-VII.5 - High relief generated with a science fiction model.

$k = 32, \alpha = 0.6$

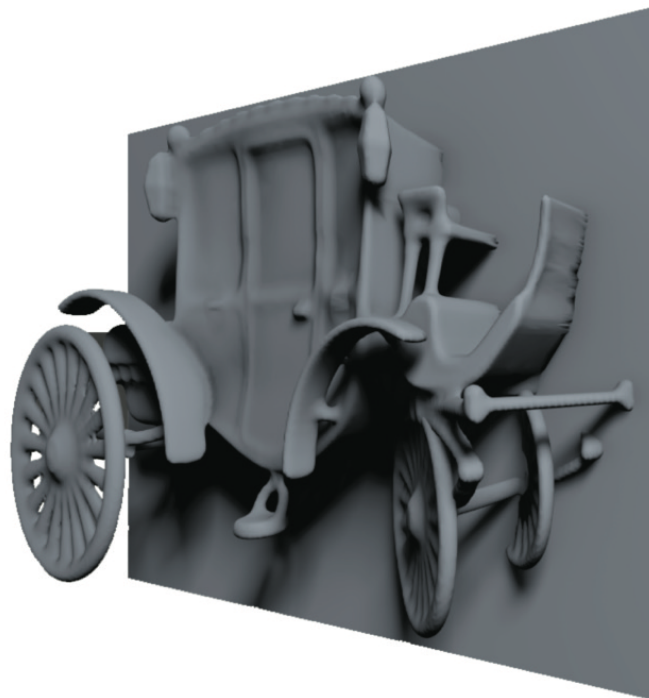


Figure A-VII.6 - High relief generated with a car model.

---

$k = 128, \alpha = 0.8$

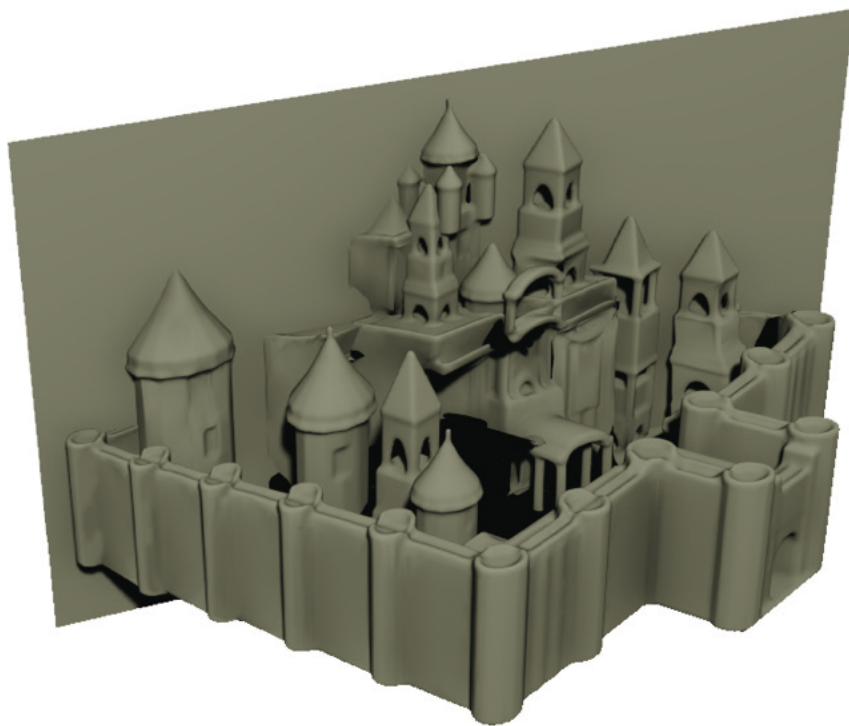
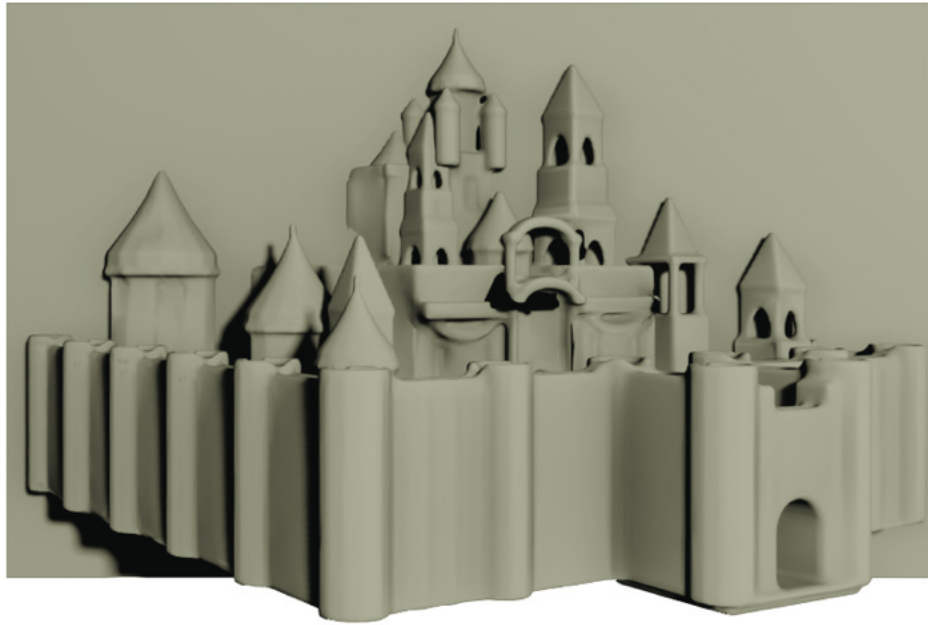


Figure A-VII.7 - High relief generated with a castle model.

$k = 64, \alpha = 0.8$

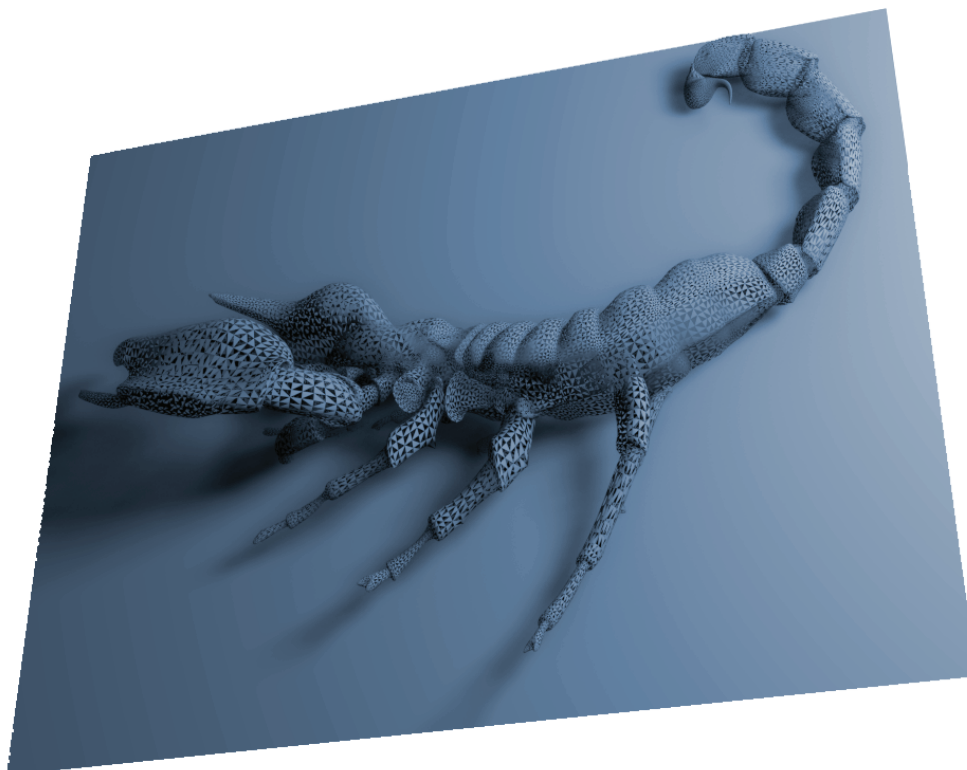


Figure A-VII.8 - High relief generated with a scorpion model.

---

$k = 32, \alpha = 0.8$

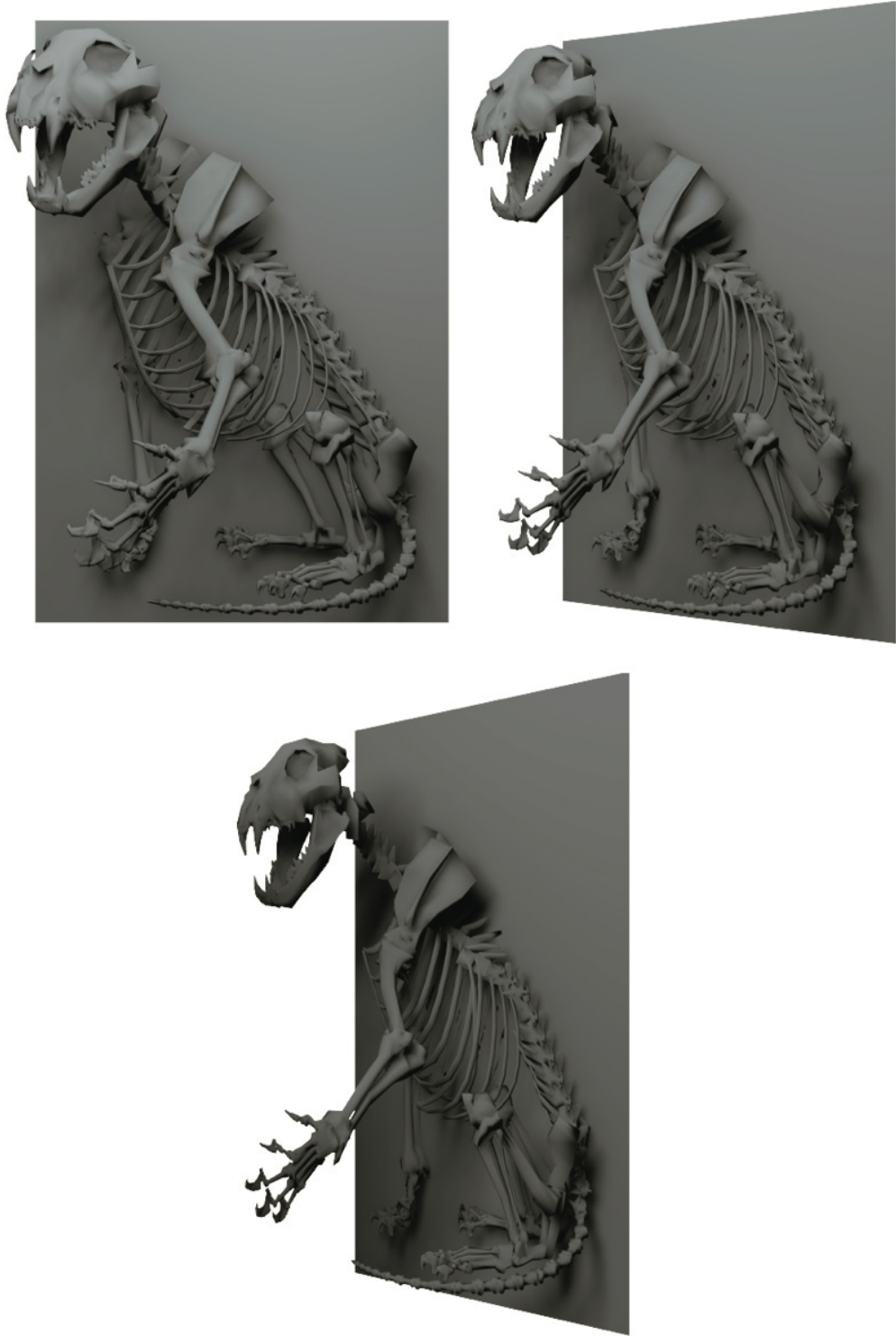


Figure A-VII.9 - High relief generated with a skeleton model.

### VIII. Averaging

Tempocodes can be revealed by averaging. With floating point numbers, contrast reduction of the input image and masking is performed without losing precision. A pixelwise average of the original tempocode video exactly yields the input target image. Smartphones can be used for averaging. The video frames are captured. By placing markers around the video, the projective distortion is removed and the rectified frames are averaged. Alternatively, a fixed camera can be used to average by integration and to reveal the target image without making any software operation.

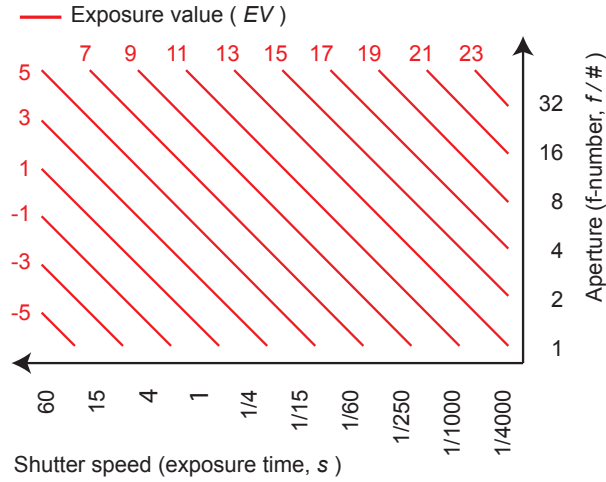


Figure A-VIII - Exposure values as a function of aperture and shutter speed [94].

The duration of temporal integration can be controlled with the shutter speed [92].

As the shutter speed is decreased for longer temporal integration, the aperture of the camera should be decreased as well in order to avoid exceeding the sensor’s holding capacity (Fig. A-VIII). *Exposure value (EV)* is a term used in photography to represent combinations of shutter speed and aperture openness (f-number) in order to obtain a given exposure. Ray [95] defines the *exposure value* as follows:

$$EV = \log_2 \frac{N^2}{t} \tag{A.1}$$

where  $N$  is the relative aperture (f-number) and  $t$  is the exposure time. Fig. A-VIII shows the exposure values for different shutter speeds and aperture values.

The messages hidden in the videos generated by our approach can be revealed by long exposure photography. However, as shown in Fig. A-VIII,  $EV$  should be lower than 10 to be able to reveal the hidden image from a tempocode having a duration longer than 1 second. We



---

performed several experiments with a Nikon D5200 camera to find the best *exposure value EV* that maps the dynamic range of the display to the dynamic range of the camera while providing as much flexibility in exposure time as possible. Modern displays have a high luminance range. By setting the screen to the lowest brightness (i.e.  $55 \pm 5 \text{ cd/m}^2$ ) and the camera ISO to 100, we found an *EV* value of 5.25 as the limit where all pixels are registered without over-exposing. Then, from in Eq. (A.2), the required aperture *f-number* for an exposure time  $t$  is:

$$N = \sqrt{t \cdot 2^{5.25}} \quad (\text{A.2})$$

For most photographic lenses, the *f-number* is limited to 32. This allows revealing the hidden image for a duration equal or shorter than 16 seconds. Longer durations can be revealed by placing a neutral density (ND) filter in front of the camera.

**Camera and screen gamma.** The presented work-flow to create videos incorporating a hidden image assume that display and camera are linear. However, usually displays raise the input signal to the power of  $\gamma$  and the cameras raise the sensor response to the power of  $\frac{1}{\gamma}$ . We linearize the display by applying to its input a power of  $\frac{1}{\gamma}$ , called gamma correction, and linearize the camera by applying to its output signal a power of  $\gamma$  before averaging. For sRGB displays, a value of  $\gamma = 2.2$  is appropriate. Alternatively a linear camera can be used.

## IX. The effect of multi-band decomposition on masking

The multi-band decomposition contributes significantly to our visual masking method. In Fig. A-IX, the effect of multi-band decomposition is shown for different contrast reduction factors  $\alpha$  and numbers of frequency bands  $k$ . The tempocode frames in the second column are generated by using the sinusoid-based composite wave with  $k = 7$  frequency bands. The target image is completely hidden in this method when the contrast reduction factor is  $\alpha = 0.4$  or lower. However, when there is no frequency band decomposition (i.e.  $k = 1$ , third column), even for the lowest contrast reduction factors, the target image is visible. Similarly, the tempocodes generated with the temporal dithering function (fourth column) can hide the target image when the contrast reduction factor is  $\alpha = 0.4$  or lower. When there is no multi-band decomposition (fifth column), the target remains visible (e.g. especially the low frequency parts).

## X. Size effect in temporal dithering

As discussed in the chapter 5, we use the parent frames generated with the random masking function as the input to our temporal dithering masking function. The masking can be achieved by directly using the contrast reduced target image as the input to our temporal

dithering masking function. However, as shown in Fig. A-X, in this scenario, the masking strongly depends on the features of the dither matrices. For instance, when the size of the dither matrix is changed, masking might not be achieved as it can be seen in Fig. A-X-1a and 4a. In Fig. A-X-1a, the high frequency content of the target image is visible and, in Fig. A-X-4a, the low frequency content of the target image is visible. Our masking algorithm requires masking all frequency bands in order to mask whole image. The success of masking depends if there are corresponding low frequency and high frequency contents in both the target image and the dither matrices. For instance, if the dither matrix has a low frequency content and the target image has a high frequency content, masking cannot be achieved. In order to prevent this, parent frames generated by our random masking function are the input to the temporal dither masking function. This enables masking all frequency channels of the target image in advance. As it can be observed in Fig. A-X 1b-4b, masking is achieved for all sizes of dither matrices.

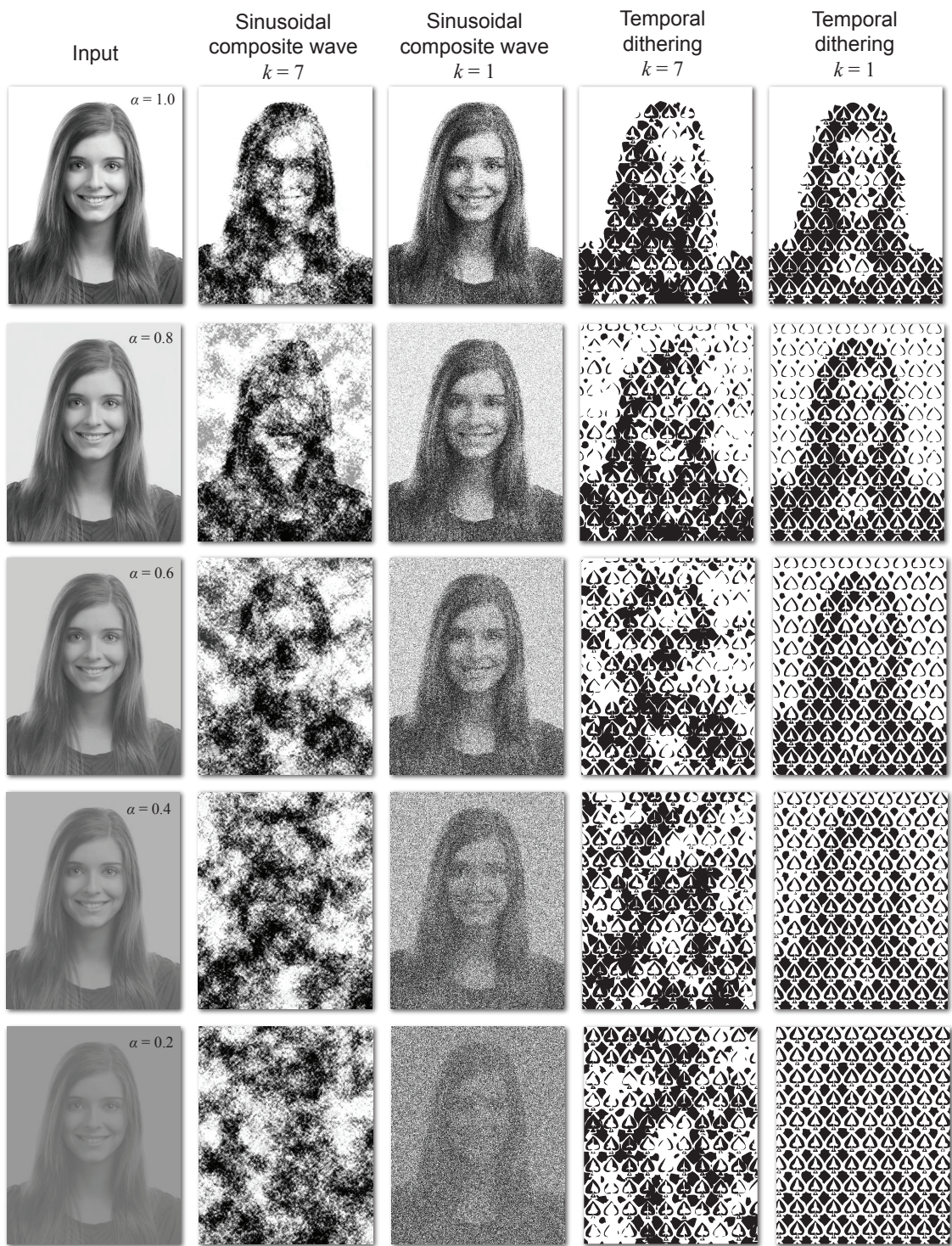


Figure A-IX - The effect of multi-band decomposition and contrast reduction on the resulting tempocode frames.

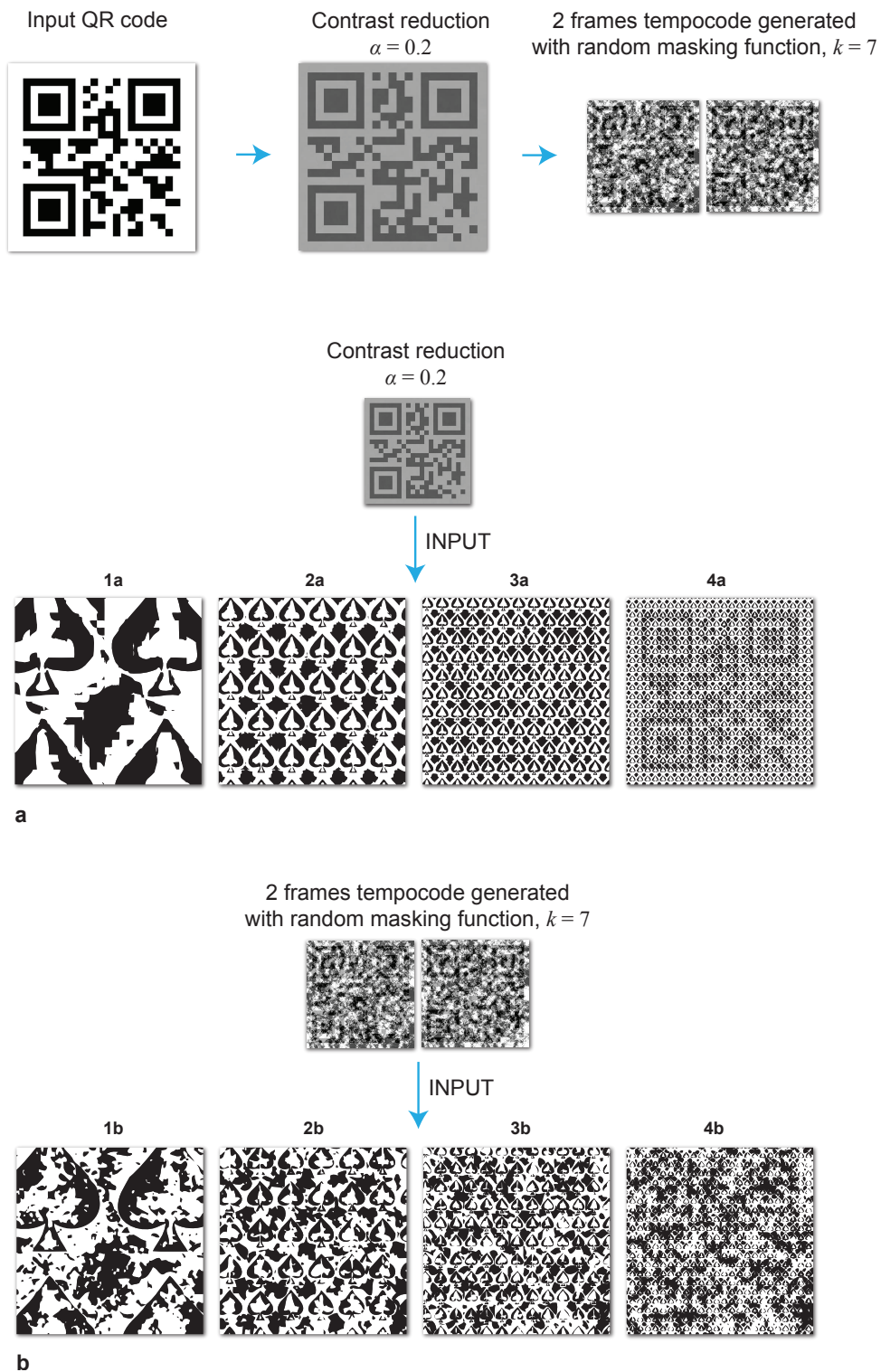


Figure A-X - a) Temporal dithering without random masking of frequency bands, for different matrix sizes. b) Temporal dithering with previously applied random masking of frequency bands, for different matrix sizes.

# Bibliography

- [1] B. Gooch and A. Gooch, *Non-photorealistic rendering*. AK Peters/CRC Press, 2001.
- [2] A. Hertzmann, “Non-photorealistic rendering and the science of art,” in *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, pp. 147–157, ACM, 2010.
- [3] R. v. Renesse, *Optical document security*. Artech House; 3rd edition, 2004.
- [4] J. P. Collomosse and P. M. Hall, “Cubist style rendering from photographs,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 4, pp. 443–453, 2003.
- [5] S. Arpa, A. Bulbul, T. Capin, and B. Ozguc, “Perceptual 3d rendering based on principles of analytical cubism,” *Computers & Graphics*, vol. 36, no. 8, pp. 991–1004, 2012.
- [6] A. Hertzmann, “Painterly rendering with curved brush strokes of multiple sizes,” in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pp. 453–460, ACM, 1998.
- [7] T. Isenberg, P. Neumann, S. Carpendale, M. C. Sousa, and J. A. Jorge, “Non-photorealistic rendering in context: an observational study,” in *Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*, pp. 115–126, ACM, 2006.
- [8] M. Mori, “The uncanny valley,” *Energy*, vol. 7, no. 4, pp. 33–35, 1970.
- [9] M. Hoffman, *Sculpture inside and out*. WW Norton & company, 1939.
- [10] V. Setlur and B. Gooch, “Is that a smile?: Gaze dependent facial expressions,” in *Proc. NPAR*, pp. 79–151, ACM, 2004.
- [11] M. Livingstone, *Vision and art: the biology of seeing*. Harry N. Abrams, 2002.
- [12] A. Oliva, A. Torralba, and P. G. Schyns, “Hybrid images,” *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 25, no. 3, pp. 527–532, 2006.
- [13] P. Didyk, T. Ritschel, E. Eisemann, K. Myszkowski, and H.-P. Seidel, “A perceptual model for disparity,” *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 30, no. 4, 2011.

## Bibliography

---

- [14] A. Hausner, "Simulating decorative mosaics," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 573–580, ACM, 2001.
- [15] H. Huang, L. Zhang, and H.-C. Zhang, "Arcimboldo-like collage using internet images," in *ACM transactions on graphics (TOG)*, vol. 30, p. 155, ACM, 2011.
- [16] V. Ostromoukhov and R. D. Hersch, "Artistic screening," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pp. 219–228, ACM, 1995.
- [17] Q. Tong, S.-H. Zhang, S.-M. Hu, and R. R. Martin, "Hidden images," in *Proc. NPAR*, pp. 27–34, 2011.
- [18] H.-K. Chu, W.-H. Hsu, N. J. Mitra, D. Cohen-Or, T.-T. Wong, and T.-Y. Lee, "Camouflage images," *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, vol. 29, no. 3, p. 51, 2010.
- [19] N. J. Mitra, H.-K. Chu, T.-Y. Lee, L. Wolf, H. Yeshurun, and D. Cohen-Or, "Emerging images," *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, vol. 28, no. 5, p. 163, 2009.
- [20] M. Papas, T. Houit, D. Nowrouzezahrai, M. Gross, and W. Jarosz, "The magic lens: Refractive steganography," *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, vol. 31, no. 6, p. 186, 2012.
- [21] M. Alexa and W. Matusik, "Reliefs as images," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 1–7, 2010.
- [22] N. J. Mitra and M. Pauly, "Shadow art," in *ACM Transactions on Graphics*, vol. 28, pp. 156–1, 2009.
- [23] B. A. Wandell, *Foundations of Vision*. Sinauer Associates, 1995.
- [24] J. A. Ferwerda, S. Pattanaik, P. Shirley, and D. P. Greenberg, "A model of visual adaptation for realistic image synthesis," in *Proc. ACM SIGGRAPH*, pp. 249–58, 1996.
- [25] G. Ward, H. Rushmeier, and C. Piatko, "A visibility matching tone reproduction operator for high dynamic range scenes," *IEEE Trans. Vis. and Comp. Graph.*, vol. 3, no. 4, pp. 291–306, 1997.
- [26] S. N. Pattanaik, J. A. Ferwerda, M. D. Fairchild, and D. P. Greenberg, "A multiscale model of adaptation and spatial vision for realistic image display," in *Proc. ACM SIGGRAPH*, pp. 287–98, 1998.
- [27] E. Durand and J. Dorsey, "Interactive tone mapping," in *Proc. EGWR*, pp. 219–230, 2000.
- [28] W. B. Thompson, P. Shirley, and J. A. Ferwerda, "A spatial post-processing algorithm for images of night scenes," *J. Graph. Tools*, vol. 7, no. 1, pp. 1–12, 2002.
- [29] S. Shlaer, "The relation between visual acuity and illumination," *J Gen Phys*, vol. 21, pp. 165–188, 1937.

- 
- [30] S. Arpa, T. Ritschel, K. Myszkowski, T. Çapın, and H.-P. Seidel, “Purkinje images: Conveying different content for different luminance adaptations in a single image,” *Computer Graphics Forum*, vol. 34, no. 1, pp. 116–126, 2015.
- [31] S. Arpa, S. Süsstrunk, and R. D. Hersch, “Sculpture paintings,” in *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering*, p. 6, ACM, 2018.
- [32] P. Cignoni, C. Montani, and R. Scopigno, “Computer-assisted generation of bas-and high-reliefs,” *Journal of graphics tools*, vol. 2, no. 3, pp. 15–28, 1997.
- [33] S. Arpa, S. Süsstrunk, and R. D. Hersch, “High reliefs from 3d scenes,” in *Computer Graphics Forum*, vol. 34, pp. 253–263, Wiley Online Library, 2015.
- [34] S. Arpa, S. Süsstrunk, and R. D. Hersch, “Revealing information by averaging,” *JOSA A*, vol. 34, no. 5, pp. 743–751, 2017.
- [35] S. E. Palmer, *“Perceiving surfaces oriented in depth” in Vision science: Photons to phenomenology*. MIT press, 1999.
- [36] T. Weyrich, J. Deng, C. Barnes, S. Rusinkiewicz, and A. Finkelstein, “Digital bas-relief from 3d scenes,” in *ACM Transactions on Graphics (TOG)*, vol. 26, pp. 32–32, ACM, 2007.
- [37] J. Kerber, M. Wang, J. Chang, J. J. Zhang, A. Belyaev, and H.-P. Seidel, “Computer assisted relief generation: A survey,” in *Computer Graphics Forum*, vol. 31, pp. 2363–2377, Wiley Online Library, 2012.
- [38] W. Song, A. Belyaev, and H.-P. Seidel, “Automatic generation of bas-reliefs from 3d shapes,” in *Shape Modeling and Applications*, pp. 211–214, IEEE, 2007.
- [39] J. Kerber, A. Tevs, A. Belyaev, R. Zayer, and H.-P. Seidel, “Feature sensitive bas relief generation,” in *Shape Modeling and Applications*, pp. 148–154, IEEE, 2009.
- [40] R. Fattal, D. Lischinski, and M. Werman, “Gradient domain high dynamic range compression,” in *ACM Transactions on Graphics (TOG)*, vol. 21, pp. 249–256, ACM, 2002.
- [41] X. Sun, P. L. Rosin, R. R. Martin, and F. C. Langbein, “Bas-relief generation using adaptive histogram equalization,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 15, no. 4, pp. 642–653, 2009.
- [42] Z. Bian and S.-M. Hu, “Preserving detailed features in digital bas-relief making,” *Computer Aided Geometric Design*, vol. 28, no. 4, pp. 245–256, 2011.
- [43] C. Schüller, D. Panozzo, and O. Sorkine-Hornung, “Appearance-mimicking surfaces,” *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH ASIA)*, vol. 33, no. 6, pp. 216:1–216:10, 2014.

## Bibliography

---

- [44] F. Campbell and J. Kulikowski, "Orientational selectivity of the human visual system," *The Journal of Physiology*, vol. 187, no. 2, p. 437, 1966.
- [45] D. M. Chandler, "Seven challenges in image quality assessment: past, present, and future research," *ISRN Signal Processing*, 2013.
- [46] R. A. Smith and D. J. Swift, "Spatial-frequency masking and birdsall's theorem," *Journal of the Optical Society of America A*, vol. 2, no. 9, pp. 1593–1599, 1985.
- [47] S. J. Daly, "Visible differences predictor: an algorithm for the assessment of image fidelity," *Proc. SPIE/IS&T Symposium on Electronic Imaging: Science and Technology*, vol. 1666, pp. 2–15, 1992.
- [48] J. A. Ferwerda, P. Shirley, S. N. Pattanaik, and D. P. Greenberg, "A model of visual masking for computer graphics," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 143–152, ACM Press/Addison-Wesley Publishing Co., 1997.
- [49] T. O. Aydin, *Human Visual System Models in Computer Graphics*. PhD thesis, Max Planck Institute for Computer Science, 2010.
- [50] D. G. Pelli and B. Farell, "Why use noise?," *Journal of the Optical Society of America A*, vol. 16, no. 3, pp. 647–653, 1999.
- [51] A. B. Watson, R. Borthwick, and M. Taylor, "Image quality and entropy masking," *Proc. SPIE Human Vision and Electronic Imaging II*, vol. 3016, pp. 2–12, 1997.
- [52] A. B. Watson and J. A. Solomon, "Model of visual contrast gain control and pattern masking," *Journal of the Optical Society of America A*, vol. 14, no. 9, pp. 2379–2391, 1997.
- [53] A. Gorea and C. W. Tyler, "New look at bloch's law for contrast," *Journal of the Optical Society of America A*, vol. 3, no. 1, pp. 52–61, 1986.
- [54] M. Kalloniatis and C. Luu, "Temporal resolution," in *Webvision* (H. Kolb, E. Fernandez, and R. Nelson, eds.), University of Utah Health Sciences Center, 1995.
- [55] F. W. Campbell and J. Robson, "Application of fourier analysis to the visibility of gratings," *The Journal of Physiology*, vol. 197, no. 3, pp. 551–566, 1968.
- [56] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "Digital image steganography: Survey and analysis of current methods," *Signal Processing*, vol. 90, no. 3, pp. 727–752, 2010.
- [57] J. Fridrich, M. Goljan, and D. Hoge, "Steganalysis of jpeg images: Breaking the f5 algorithm," *Proc. Information Hiding*, pp. 310–323, 2003.
- [58] Z. Li, X. Chen, X. Pan, and X. Zeng, "Lossless data hiding scheme based on adjacent pixel difference," *Proc. International Conference on Computer Engineering and Technology*, vol. 1, pp. 588–592, 2009.



- [59] P. Tsai, Y.-C. Hu, and H.-L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal Processing*, vol. 89, no. 6, pp. 1129–1143, 2009.
- [60] X. Li and J. Wang, "A steganographic method based upon jpeg and particle swarm optimization algorithm," *Information Sciences*, vol. 177, no. 15, pp. 3099–3109, 2007.
- [61] A. Hashad, A. S. Madani, A. E. M. A. Wahdan, *et al.*, "A robust steganography technique using discrete cosine transform insertion," *Proc. IEEE International Conference on Information and Communications Technology*, pp. 255–264, 2005.
- [62] K. Raja, C. Chowdary, K. Venugopal, and L. Patnaik, "A secure image steganography using lsb, dct and compression techniques on raw images," *Proc. IEEE International Conference on Intelligent Sensing and Information Processing*, pp. 170–176, 2005.
- [63] R. T. McKeon, "Strange fourier steganography in movies," *Proc. IEEE International Conference on Electro/Information Technology*, pp. 178–182, 2007.
- [64] P. Sallee, "Model-based steganography," in *Digital Watermarking*, pp. 154–167, Springer, 2004.
- [65] H. Hirohisa, "A data embedding method using bpcs principle with new complexity measures," *Proc. Pacific Rim Workshop on Digital Steganography*, pp. 30–47, 2002.
- [66] P. Wayner, *Disappearing cryptography: information hiding: steganography & watermarking*. Morgan Kaufmann, 2009.
- [67] G. C. Langelaar, I. Setyawan, and R. L. Legendijk, "Watermarking digital image and video data. a state-of-the-art overview," *IEEE Signal Processing Magazine*, 17 (5), 2000.
- [68] A. Khan, A. Siddiqa, S. Munib, and S. A. Malik, "A recent survey of reversible watermarking techniques," *Information Sciences*, vol. 279, pp. 251–272, 2014.
- [69] M. Arsalan, S. A. Malik, and A. Khan, "Intelligent reversible watermarking in integer wavelet domain for medical images," *Journal of Systems and Software*, vol. 85, no. 4, pp. 883–894, 2012.
- [70] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-lsb data embedding," *Image Processing, IEEE Transactions on*, vol. 14, no. 2, pp. 253–266, 2005.
- [71] G. Xuan, C. Yang, Y. Zhen, Y. Q. Shi, and Z. Ni, "Reversible data hiding using integer wavelet transform and companding technique," *Proc. Digital Watermarking*, pp. 115–124, 2005.
- [72] C.-C. Lin, W.-L. Tai, and C.-C. Chang, "Multilevel reversible data hiding based on histogram modification of difference images," *Pattern Recognition*, vol. 41, no. 12, pp. 3582–3591, 2008.

## Bibliography

---

- [73] A. Khan, S. A. Malik, *et al.*, “A high capacity reversible watermarking approach for authenticating images: Exploiting down-sampling, histogram processing, and block selection,” *Information Sciences*, vol. 256, pp. 162–183, 2014.
- [74] M. Alexa, “Differential coordinates for local mesh morphing and deformation,” *The Visual Computer*, vol. 19, no. 2, pp. 105–114, 2003.
- [75] Z. Karni and C. Gotsman, “Spectral compression of mesh geometry,” in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 279–286, ACM Press/Addison-Wesley Publishing Co., 2000.
- [76] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel, “Laplacian surface editing,” in *Proceedings of the 2004 Eurographics, ACM SIGGRAPH symposium on Geometry processing*, pp. 175–184, 2004.
- [77] P. Murray and L. Murray, *A dictionary of art and artists*. Penguin Books, 1972.
- [78] M. Hoffman, *Sculpture inside and out*. Bonanza Books, New York, 1939.
- [79] J. D’Amelio and S. Hohausser, *Perspective drawing handbook*. Courier Dover Publications, 2004.
- [80] R. Arnheim, *Art and visual perception: A psychology of the creative eye*. Univ of California Press, 1974.
- [81] J. MacQueen *et al.*, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, p. 14, California, USA, 1967.
- [82] O. Sorkine, “Laplacian mesh processing, <http://www.cs.berkeley.edu/~jrs/meshpapers/Sorkine.pdf>,” *Eurographics STAR - State of The Art Report*, pp. 53–70, 2005.
- [83] P. N. Belhumeur, D. J. Kriegman, and A. L. Yuille, “The bas-relief ambiguity,” *International Journal of Computer Vision*, vol. 35, no. 1, pp. 33–44, 1999.
- [84] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, “Color transfer between images,” *IEEE Computer graphics and applications*, vol. 21, no. 5, pp. 34–41, 2001.
- [85] P. Su and R. L. S. Drysdale, “A comparison of sequential delaunay triangulation algorithms,” *Computational Geometry*, vol. 7, no. 5-6, pp. 361–385, 1997.
- [86] D. Summers, *Texturing: Concepts and Techniques*. Charles River Media graphics, Charles River Media, 2004.
- [87] P. J. Burt and E. H. Adelson, “The laplacian pyramid as a compact image code,” *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, 1983.
- [88] M. N. Do and M. Vetterli, “Framing pyramids,” *IEEE Transactions on Signal Processing*, vol. 51, no. 9, pp. 2329–2342, 2003.

- [89] V. Ostromoukhov and R. D. Hersch, "Multi-color and artistic dithering," *Proc. 26th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 425–432, 1999.
- [90] C. Hains, S. Wang, and K. Knox, *Digital Color Halftones, Chapter 6 in Digital Color Imaging Handbook*. Boca Raton, FL: CRC, 2003.
- [91] R. D. Hersch and B. Wittwer, "Method and computing system for creating and displaying images with animated microstructures," Nov. 24 2009. US Patent 7,623,739.
- [92] E. Allen and S. Triantaphillidou, *The Manual of Photography and Digital Imaging*. CRC Press, 2012.
- [93] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," *Proc. Signals, Thirty-Seventh Asilomar Conference on Systems and Computers*, vol. 2, pp. 1398–1402, 2003.
- [94] Canon, "Camera settings: shooting modes," 2015.
- [95] S. Ray, "Camera exposure determination," in *Manual of Photography* (R. Jacobson, S. Ray, G. G. Attridge, and N. Axford, eds.), Taylor & Francis, 2000.





# Sami Arpa

✉ msamiarpa@gmail.com  
🌐 ivrlwww.epfl.ch/~arpa/

☎ +41 78 908 7502  
🌐 www.linkedin.com/in/samiarpa/

Chemin de la Joliette 4  
1006 Lausanne / SWITZERLAND

## EDUCATION

- 2013-2018  
EPFL  
Computer Science  
PhD  
Lausanne / SWITZERLAND
- 2010-2012  
Bilkent University  
Computer Engineering  
MS - Full scholarship  
Ankara / TURKEY
- 2005-2010  
Bilkent University  
Computer Engineering  
BS - Full scholarship  
Ankara / TURKEY

## PROFESSIONAL SKILLS

UX Designer (+10 years), C/C++ (advanced), Java (advanced), Matlab (advanced), OpenGL (advanced), GLSL, OpenCV, PHP, MySQL, Python, Javascript, Maya, 3DsMax, After Effects, Premiere, Photoshop, InDesign, Illustrator

## SELECTED PUBLICATIONS

- Sami Arpa, Sabine Süsstrunk, Roger D. Hersch  
Sculpture Paintings  
Proceedings of Expressive 18, 2018
- Sami Arpa, Sabine Süsstrunk, Roger D. Hersch  
Revealing Information by Averaging  
Journal of the Optical Society of America A, 2017
- Sami Arpa, Sabine Süsstrunk, Roger D. Hersch  
High Reliefs from 3D Scenes  
Computer Graphics Forum 34(2), EUROGRAPHICS, 2015
- Sami Arpa, Tobias Ritschel, K. Myszkowski, T. Çapın and H. P. Seidel  
Purkinje Images: Conveying Different Content for  
Different Luminance Adaptations in a Single Image  
Comp. Graph Forum 33(2), 2014
- Sami Arpa, Abdullah Bulbul, Tolga Capin, Bulent Ozguc  
Perceptual 3D rendering based on principles of analytical cubism  
Computers & Graphics 36.8, 2012

## PATENTS

- Sami Arpa, Sabine Süsstrunk, Roger D. Hersch  
2D-3D Combined Sculpture Paintings  
US Patent (filed), 2018
- Sami Arpa, Sabine Süsstrunk, Roger D. Hersch  
Synthetic Electronic Video Containing A Hidden Image  
US Patent (filed), 2018

## EXPERIENCE

- 2017-present  
LARGO FILMS  
CEO / Co-founder  
Next generation storytelling  
Lausanne / SWITZERLAND
- 2013-present  
EPFL  
Research Assistant  
Computer Graphics & Vision  
Lausanne / SWITZERLAND
- 2012-2013  
3Dios  
Co-founder  
Startup for 3D stereo moviemaking  
Ankara / TURKEY
- 2012-2013  
MAX PLANCK INT.  
Research Assistant  
Visual Perception & Graphics  
Saarbrücken / GERMANY
- 2010-2012  
BILKENT UNIV.  
Research Assistant  
Aesthetics & Graphics  
Ankara / TURKEY

## EXPERIENCE IN CINEMA

- 2015-present  
OUCHY FILM AWARDS  
President / Founder  
Lausanne / SWITZERLAND
- 2015  
- L'évasion -  
Director / Co-producer  
A short film shot in Lausanne.  
17 min.
- 2014  
- Les bruits des pas -  
Director / Producer  
A short film shot in Lausanne.  
15 min.

## LANGUAGES

Turkish - Native    English - C1    French - B1

## AWARDS - HONORS

- 50.000 Euro - Award for technological enterprise by Turkish Ministry of Science and Technology, June 2012.
- Microsoft Visual Studio 2010 Certificate of Recognition, 24th May 2010
- Bilkent University Graduation with Honor Degree, June 2010
- Bilkent University Full Scholarship Award upon the success in entrance exam (Bachelor Degree), August 2005
- Ranked in top 500 among 1,700,000 participants in the national university entrance exam in Turkey, August 2005

